AD-772 433

NAVAL RESEARCH LOGISTICS QUARTERLY. VOLUME 18, NUMBER 1

Office of Naval Research
Arlington, Virginia

March 1971

Best
Available
Copy

NAVAL RESEARCH LOGISTICS QUARTERLY

MARCH 1971
VOL. 18, NO. 1

VOL 17 - NO. 4 - AD 772 432

OFFICE OF NAVAL RESEARCH

NAVSO P-1278

NAVAL RESEARCH LOGISTICS QUARTERLY

The Naval Research Logistics Quarterly is devoted to the dissemination of scientific information in logistics and will publish research and expository papers, including those in certain areas of mathematics, statistics, and economics, relevant to the over-all effort to improve the efficiency and effectiveness of logistics operations.

Information for Contributors is indicated on inside back cover.

The Naval Research Logistics Quarterly is published by the Office of Naval Research in the months of March, June, September, and December and can be purchased from the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. Subscription Price: $5.50 a year in the U.S. and Canada, $7.00 elsewhere. Cost of individual issues may be obtained from the Superintendent of Documents.

TREE-SEARCH ALGORITHMS FOR QUADRATIC ASSIGNMENT PROBLEMS*

J. F. Pierce

Management Decisions Development Corp.
Cincinnati, Ohio

and

W. B. Crowston

Sloan School of Management
Massachusetts Institute of Technology
Cambridge, Massachusetts

ABSTRACT

Problems having the mathematical structure of a quadratic assignment problem are found in a diversity of contexts: by the economist in assigning a number of plants or indivisible operations to a number of different geographical locations; by the architect or industrial engineer in laying out activities, offices, or departments in a building; by the human engineer in arranging the indicators and controls in an operators control room; by the electronics engineer in laying out components on a backboard; by the computer systems engineer in arranging information in drum and disc storage; by the production scheduler in sequencing work through a production facility; and so on.

In this paper we discuss several types of algorithms for solving such problems, presenting a unifying framework for some of the existing algorithms, and describing some new algorithms. All of the algorithms discussed proceed first to a feasible solution and then to better and better feasible solutions, until ultimately one is discovered which is shown to be optimal.

I. INTRODUCTION

The quadratic assignment problem is one which arises in a diversity of contexts and has been investigated by a number of reseachers. Formally, the problem may be stated simply as follows: given n^4 cost coefficients S_{ijkq}, $(i, j, k, q = 1, 2, 3, \ldots, n)$ determine values of the n^2 variables $x_{ij} (i, j = 1, 2, 3, \ldots, n)$ so as to:

Minimize

$$Z = \sum_{i,j} \sum_{k,q} S_{ijkq} x_{ij} x_{kq} \tag{1}$$

Subject to:

$$\sum_{i=1} x_{ij} = 1 \qquad j = 1, 2, 3, \ldots, n, \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, 2, 3, \ldots, n, \tag{3}$$

*The authors gratefully acknowledge the support of this research received from the Boeing Research Project and from the National Science Foundation under grant No. 70985.

and

(4) $$x_{ij} = 0, 1 \qquad i, j = 1, 2, \ldots, n.$$

Historically its name derives from the fact that mathematically its structure is identical to that of the classical linear assignment problem concerning the assignment of n indivisible entities to each of the n mutually exclusive classes, one entity per class, except that in the present case the objective function (1) contains terms which are quadratic in the decision variables.

Commencing in the field of economics, Koopmans and Beckmann [15] identified this name with the structure of problems which concern the assignment of n indivisible plants to n locations. Suppose the cost of establishing and operating plant i at location j plus the cost of supplying prespecified product demand to customers from this location is c_{ij}, $i, j = 1, 2, \ldots, n$, these costs being independent of other plant-location assignments. Also suppose that between plants i and k there is a commodity flow of f_{ik} units (e.g., weight) which is independent of plant location, and that the cost per unit flow between locations j and q is d_{jq}, independent of plant assignments. Then in this context (1) becomes:

(5) $$Z = \sum_{i,j} c_{ij} x_{ij} + \sum_{i,j} \sum_{k,q} f_{ik} d_{jq} x_{ij} x_{kq}$$

where

$$S_{ijkq} = \begin{Bmatrix} f_{ik} \cdot d_{jq}, & \text{if } i \neq k \text{ or } i \neq q \\ c_{ij} + f_{ii} d_{jj}, & \text{if } i = k \text{ and } j = q \end{Bmatrix}.$$

As a generalization to this assignment problem, Lawler [18] discusses the multicommodity case in which there is a flow f^t_{ik} for each commodity t and a cost per unit flow between locations j and q of d^t_{jq}. As another generalization, Graves and Whinston [12] point out the possibility in this model of a cost component π_{ijkq} that depends on a pair of assignments, such as might be illustrated by the cost of laying a pipe line between two plants. Combining these we thus have the more general cost expression:

(6) $$Z = \sum_{i,j} c_{ij} x_{ij} + \sum_{i,j} \sum_{k,q} \pi_{ijkq} x_{ij} x_{kq} + \sum_{i,j} \sum_{k,q} \sum_{t} f^t_{ik} d^t_{jq} x_{ij} x_{kq}$$

where

$$S_{ijkq} = \begin{Bmatrix} \pi_{ijkq} + \sum_t f^t_{ik} d_{jq} & \text{if } i \neq k \text{ or } j \neq q \\ c_{ij} + f_{ii} d^t_{jj} & \text{if } i = k \text{ and } j = q \end{Bmatrix}.$$

In the event there are no inter-plant flows $f^t_{ik} = o$ for all i, k, t, and the problem in (5) reduces to the linear assignment problem. When $c_{ij} = 0$ for all i, j and

$$f_{ik} = \begin{cases} 1 & k = i + 1, \; i < n \\ 1 & i = n, \; k = 1 \\ 0 & \text{otherwise} \end{cases}$$

the problem reduces to the traveling salesman problem [18].

At a more micro-economic level this problem arises in the context of locating department or offices within a plant or store to minimize the cost of transporting product, the total distance walked, or some similar measure [1, 24, 31, 32]. At a still more micro level it is the problem of locating operator dials and indicators on a display and control panel. In other contexts (1) – (4) is the problem of minimizing

"latency" in magnetic drum or disc storage computers [19], minimizing total wire length in the placement of electronic components in assemblies [2, 7, 30], or minimizing total flow time or total variable production and inventory carrying cost in various production sequencing problems [22].

In some contexts there may be constraints applicable to the problem which are not represented in the statement as embodied in (1) – (4). For example, there may be a restriction that plant i not be located at j, or a restriction that plants i and k be not more than distance, d, apart, or that i and k be closer than d. All single and pairwise constraints of this kind are readily accommodated in (1) – (4) by setting $s_{ijkq} = M$, $M \to \infty$. However, more difficult to include are constraints involving three or more assignments unless it is possible to derive an equivalent set of pairwise constraints. While the algorithms to be discussed can be adapted for such cases the resulting algorithms may not be as efficient.

From a problem-solving point of view there may in practice be fewer than n plants, $m < n$, but with no loss of generality we may assume $m = n$ by introducing dummy plants $m+1, m+2, \ldots, n$ with $c_{ij} = 0$ and $f_{ik} = 0$ for all $i, k > m$. Also it is noted that, stated in terms of a plant i and its location $\ell(i)$, problem (1) – (4) and its variations is the problem of finding a permutation $\{\ell(1), \ell(2), \ell(3), \ldots, \ell(n)\}$ of the integers $\{1, 2, 3, \ldots, n\}$ so as to minimize:

$$Z = \sum_{i,k} s_{ik\ell(i)\ell(k)}.$$

This representation will sometimes be used in the following discussion.

For solving quadratic assignment problems a number of procedures of both the reliable* and the unreliable type have been reported in the literature. Reliable procedures for determining optimal solutions with objective function (5) have been presented by Gilmore [10] and Lawler [15], and for the symmetric case† of (5) by Land [16] and Gavett and Plyter [8]. For the problem with the general objective function (6) a reliable algorithm has been given by Lawler [18]. On the other hand unreliable procedures have been reported for various quadratic assignment problems by Armour and Buffa [1], Gaschutz and Ahrens [7], Gilmore [19], Graves and Whinston [12], Hillier [13], Hillier and Connors [14], Nugent, Vollman, and Ruml [23], Pegels [24], Steinberg [30], Whitehead and Eldors [32] and by Wimmert‡ [33]. An interesting experimental comparison of a number of these latter procedures is presented in the paper by Nugent, et al. [23].

From a computational point of view, the present status can perhaps be succinctly summarized as follows. Existing reliable algorithms can essentially be classified into three groups: the integer programming approach of Lawler [18]; the semi-enumerative procedures of Lawler [18] and Gilmore [10]; and the semi-enumerative approaches of Gavett and Plyter [8] and Land [16]. With presently available integer programming algorithms the first approach is impractical even for small problems, in light of the size of the programming problem which results. For the second group we know of no actual computational experience with the algorithms, but as stated by Gilmore [10] his reliable algorithms are "probably not computationally feasible for n much larger than 15." For the third group Gavett and Plyter [8] report that with their algorithm as programmed in Fortran on an IBM 7044, a problem with $n = 7$ required 11 minutes and one with $n = 8$, 42 minutes. In short, in the words

*By a reliable problem solving procedure we shall mean one which, if carried through to completion, guarantees the discovery of an optimal solution.

†A symmetric distance matrix $d_{jq} = d_{qj}$ for all j and q allows the flow between activities to be summed, $f'_{ik} = f_{ik} + f_{ki}$, thereby possibly simplifying the problem solving process.

‡The algorithm of Wimmert was originally presented as yielding optimal solutions but was subsequently shown by Conway and Maxwell [5] to yield suboptimal solutions.

of Nugent, et al., "one is forced to conclude that no computationally feasible optimal-producing procedure exists at present. Interest must focus on suboptimal procedures."

In the present paper, we redirect attention back to reliable procedures for solving quadratic assignment problems. The methods to be considered are those which equivalently been referred to as branch and bound procedures [20], back-track programming procedures [11], implicit enumeration procedures [9], reliable heuristic programming procedures [25], and others. Essentially these are the types of methods that were used in the algorithms of Gavett and Plyter [8], Gilmore [10], Land [16], and Lawler [18]. In the following sections we present a unified framework in which to compare the existing algorithms, and discuss some alternative search strategies and other means by which it may be possible to devise more efficient procedures.

Before turning to the algorithms in detail, however, we shall comment briefly on the nature of the methods to be considered and the reasons for our inclination toward them. The most common name for the procedures to be investigated is "branch and bound," the name given to the ideas employed by Little et al. [20] in their algorithms for solving the traveling salesman problem. The "branch" notion stems from the fact that in terms of a tree of alternate potential solutions to the problem the procedure is continually concerned with choosing a next branch of the tree to elaborate and evaluate. The "bound" term denotes their emphasis on, and effective use of, means of bounding the value of the objective function at each node in the tree, both for eliminating dominated paths and for selecting a next branch for elaboration and evaluation.

Perhaps the essence of the procedures to be considered is most succinctly captured in the meaning given to "combinatorial programming" by its authors Rossman and Twery [26]. By combinatorial programming we mean procedures developed on the basis of two principal concepts: the use of a controlled enumerative techinque for (implicitly) considering all potential solutions: and the elimination from explicit consideration of particular potential solutions which are known from dominance, bounding and feasibility considerations to be unacceptable. All of the equivalent terms for these methods will be used interchangeably throughout.

As will become apparent many of the feasibility, dominance, and bounding considerations presented in the following sections are also applicable in other combinatorial programming algorithms as well as in other types of problem-solving procedures. In the following sections, attention will be focused on combinatorial programming algorithms in which problem-solving proceeds first to the discovery of a feasible solution and then to successively better feasible solutions until ultimately one is discovered which is shown to be optimal. We direct attention to these procedures principally because they have the following three desirable attributes.

First, with such procedures there is a possibility of obtaining usable solutions and terminating problem-solving prior to the ultimate completion of the problem-solving process. This feature is obviously important for quadratic assignment problems.

Second, these procedures exploit in an efficient manner information that is available beforehand pertaining to the value of an optimal solution, as is always the case for instance when a feasible solution is known from experience or has been derived with the aid of a heuristic (unreliable) procedure. That is, since in the procedures to be investigated subsequent search is always directed toward solutions with a value better than the best known so far and will terminate if a solution is discovered attaining a known lower bound, use of a priori knowledge of upper and/or lower bounds serves to reduce the region that need be searched. Therefore, in contexts where good heuristic procedures are available, for example, a system of problem solving procedures may prove advantageous in which the reliable,

direct algorithm is employed as an adjunct to the heuristic procedures, an adjunct to be employed when in a given instance the economics of the problem and problem-solving effort together with environmental considerations warrant the added search for a solution better than that yielded by the heuristic procedures.

Thirdly, these algorithms are attractive in that with slight modification they can be employed to find not only *an* optimal solution, but *all* optimal solutions, or a specified number of most preferred solutions, or all solutions having a value within a specified interval of the optimal value, and so on. Such possibilities may be of interest in contexts in which there are attributes of the problem of importance which are not directly represented in the model of the problem being solved.

II. SINGLE-ASSIGNMENT ALGORITHMS

As noted earlier, a first principle of combinatorial programming is the use of a controlled enumeration procedure for systematically considering, at least implicitly, all potential solutions. For quadratic assignment problems there are at least two general procedures: one based on the systematic consideration of single assignments, x_{ij}, and one based on the systematic consideration of pairs of assignments, $x_{ij}x_{kq}$. Both types have appeared in reliable algorithms to date, Gilmore [10] and Lawler [18] using the former, and Land [16] and Gavett and Plyter [8] using the latter. In this section we consider algorithms of the former type.

A property of a feasible solution to the problem (2)–(4) is that with the variables x_{ij} arranged in an $n \times n$ matrix $\mathbf{X} = \|x_{ij}\|$ there exists exactly one variable in each row and column of the assignment matrix $\mathbf{X}$ having unit value. To satisfy the requirements for considering all potential solutions, we therefore need a controlled enumeration procedure for generating all possible ways of selecting one element from each row and column of $\mathbf{X}$. One possible procedure, for example, is to successively select elements from successive rows of $\mathbf{X}$ and to select within a given row the first element (when scanned from left to right, say) which will result neither in a nonfeasible solution nor in a solution already generated. Ultimately upon making a selection from row n and hence completing the specification of a solution, the procedure backs up to row $n-1$, selects the next admissible element and steps forward to row n again. The results may be represented in a tree structure with the ith level of nodes representing the permissible assignments for plant i, $\ell(i)$, in the permutation $\{\ell(1), \ell(2), \ldots, \ell(n)\}$ as shown in Figure 1. Note that each path in this tree represents a feasible solution to our problem.

The procedure described above would thus elaborate the tree shown from left to right. Enumeration may thus be equivalently viewed as entailing the successive row by row selection of an element from matrix $\mathbf{X}$ to include in the permutation $\theta = \{\ell(1), \ell(2), \ldots, \ell(n)\}$ or as entailing the successive level by level selection of branches in a tree (one branch per level) until a terminal node is reached at

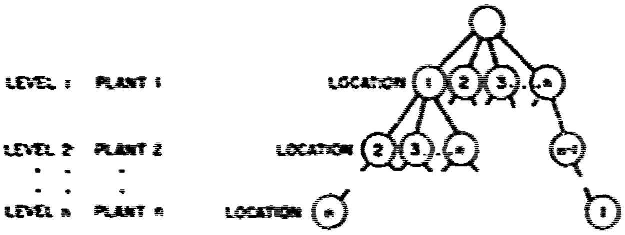


FIGURE 1. Illustrative tree with each level representing a unique plant.

level n. Upon reaching a terminal node, the corresponding assignment is evaluated and the tree-evaluation process backtracks to the lowest node on the path for which all branches have not been elaborated, selects the next and resumes. When the process has backtracked to the origin node and all its branches have been enumerated, generation and hence problem solving is complete.

In addition to this illustrative enumeration scheme there are many others for systematically selecting an element from each row and column of $\mathbf{X}$. For example, if we interchange the words "row" and "column" in the cited procedure, we have a tree with levels corresponding to locations rather than plants, as shown in Figure 2.

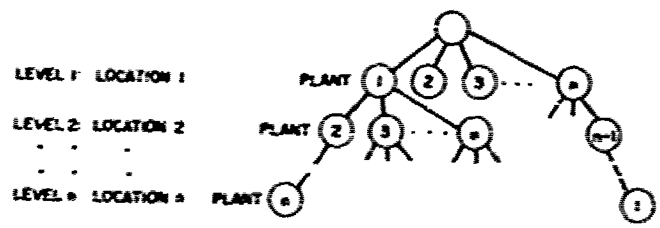


FIGURE 2. Illustrative tree with each level representing a unique location.

If in the process of exhaustive enumeration, it becomes known with certainty for a particular node assignment, that all paths which pass through this node represent potential solutions which are nonfeasible or are dominated by a feasible solution already discovered, then the enumeration and evaluation of all branches emanating from this node can be eliminated without impairing the reliability of the problem-solving procedure. Let us consider possibilities for reducing search based on dominance considerations. For any two feasible solutions, θ_i and θ_j, θ_i dominates θ_j if $Z_{\theta_i} \leq Z_{\theta_j}$. If in an exhaustive procedure θ_i denotes the ith feasible solution discovered then, in general, $Z_{\theta_i} \gtreqless Z_{\theta_{i+1}}$. Through dominance consideration we seek to reduce enumeration and evaluation of feasible solutions to a subset in which

$$Z_{\theta_1} > Z_{\theta_2} > \ldots > Z_{\theta_{(u)}},$$

where $Z_{\theta_{(u)}}$ is an optimal solution.* In effect, this is accomplished by affixing to the problem throughout the search process a constraint of the form $Z_{\theta_j} < Z_{\hat\theta}$ where θ_j is the best solution discovered so far. In essence, the optimization problem is hereby transformed into a sequence of u feasibility problems for purposes of problem-solving. To implement this type of consideration a lower bound B is developed on Z_{θ_j} at each node in the tree for all θ_j whose paths pass through the given node, that is, $B \leq Z_{\theta_j}$ for all θ_j; if $B \geq Z_{\hat\theta}$ then no branches emanating from the node need be explicitly considered.

For the quadratic assignment problem, these bounds can be determined in a number of ways. Suppose we have arrived at a node on level r of a tree of the problem ($r = 0, 1, \ldots, n-1$) having made assignments $(i, \ell(i))$ and we now wish to choose a next assignment, x_{ij}. Let a^*_{ij} be a lower bound on the sum

$$S_{ijij} + \sum_{k \in I} S_{k\ell(k)ij} + \frac{1}{2} \sum_{p \notin I} S_{pq\cdot pq}$$

*It is possible that more than one optimal solution could exist, e.g., $Z_{\theta_{(u)}} = Z_{\theta_{(u+1)}} = \ldots = Z_{\theta_{(v)}}$. If all such solutions are desired then the set of feasible solutions would be $Z_{\theta_1} > Z_{\theta_2} > \ldots Z_{\theta_{(v)}}$.

where I is the set of assigned plants i. Since the first two terms are known exactly, we have

$$a_{ij}^{r} = S_{(ij)} + \sum_{k \in I} S_{(jk)(\ldots)} + \frac{1}{2} \hat{a}_{ij}^{r} \tag{7}$$

where $\hat{a}_{ij}^{r}$ is a lower bound on the sum of $(n-r)$ terms.

$$\sum_{p \notin I} S_{(\ldots)(p\ldots)}$$

As noted by Lawler a minimum bound $\hat{a}_{ij}^{r}$ can be obtained by solving the linear assignment problem of dimension $(n-r)$. In the special case of (5) where $S_{ijkq} = (f_{ik} d_{jq} + f_{ki} d_{qj})$, both Lawler and Gilmore point out that a lower bound is more easily obtained by matching the largest value of f_{ik} with the smallest d_{jq}, the next to largest f_{ik} with the second smallest d_{jq} and so on. For the symmetric case of $d_{jq} = d_{qj}$ and $t = 1$, $S_{ijkq} = (f_{ik} + f_{ki}) d_{jq}$ so that the sum of the products of these values actually gives a minimum value. This applies equally to the case

$$S_{ijkq} = \sum_{t} (f_{ik}^{t} d_{jq}^{t} + f_{ki}^{t} d_{qj}^{t}).$$

Thus by determining an appropriate bound $\hat{a}_{ij}^{r}$ a value can be obtained for each unassigned plant i and location j remaining at level r. Let $\mathbf{A}^{r}$ denote the resulting $(n-r)$ matrix $\|a_{ij}^{r}\|$. If we denote by

$$Z_r^{*} = \text{Min.} \left\{ \sum_{ij} a_{ij}^{r} x_{ij} \right\}$$

the value of an optimal solution to a linear assignment problem defined by $\mathbf{A}^{r}$, then a lower bound B^{r} on all feasible solutions whose path passes through the node is

$$B^{r} = Z_r^{*} + \sum_{i,\, k \in I} S_{(\ldots)(\ldots)} \tag{8}$$

Thus if $B^{r} \geq Z_0$ then the search process can be backtraced immediately without considering any of the branches emanating from the node.

An alternative to this bound which requires less computation (but is also less stringent) is suggested by Gilmore for the Koopmans-Beckmann problem which he investigates. With the objective function

$$Z = \sum_{k \in I} c_{(\ldots)} + \sum_{i,\, k \in I} f_{ik} d_{(\ldots)(\ldots)}$$

he suggests at level r the bound

$$B_r = \sum_{k \in I} c_{(\ldots)} + \sum_{i,\, k \in I} f_{ik} d_{(\ldots)(\ldots)} + S_1 + S_2 + S_3, \tag{9}$$

where S_1 is the value of an optimal solution to the $(n-r)$ dimensional assignment problem defined by $\{c_{ij}\}\, i \neq I$ and $j \neq r(i)$ for any $i \in I$.

$$S_2 = \sum_{i,\, k \in I} (f_{ik} \cdot \dot{d}_{\ldots} + \dot{f}_{ki} \cdot \dot{d}_{\ldots})$$

$$j \neq r(q),\ q \in I.$$

where the largest element $\hat{f}_{ik}$ is matched with the smallest element $\hat{d}_{\ell(w)}$, and so on, and likewise for $\hat{f}_{i0}$ and $\hat{d}_{j,\ell(w)}$, and finally where

$$S_j = \sum_{\substack{i,k \neq l \\ j,g \neq \ell(w)}} f_{ik}^{*} d_{jg}^{*} \qquad \text{any } w \in l,$$

where the largest element of f_{ik}^{*} is paired with the smallest element d_{jg}^{*}, etc.

Let us now return to the discussion of controlled enumeration procedures. The controlled enumeration procedures mentioned previously are data-independent with respect to the order in which potential solutions are investigated: for every problem having the same number n of plants the order is identical regardless of the characteristics of the particular problem being solved.* In such procedures little problem-solving time is invested in determining a next branch in the tree for investigation and (at least in the procedures discussed)† in keeping track of the part of the tree investigated so far. Perhaps, however, more efficient combinatorial programs may result by expending additional time on these functions and making the ordering of search more dependent on the particular features of the problem being solved.

There are at least two basic search patterns of a general data-dependent nature wherein at each point in the search process a branch is selected for elaboration to the next level which has associated with it a most preferred value of a measure B'. A natural characteristic to employ as a measure B', for instance, is a lower bound on the total cost B_T of all potential feasible solutions passing through the node.

In the "flooding" type pattern a branch is always selected from among *all* branches in the current tree requiring elaboration to a next level. In the second type pattern, search is directed towards the enumeration of complete paths so that in turn one branch is selected at level 1, then one at level 2, etc.: at level j all branches emanating from the single node selected at level $j-1$ are evaluated in terms of the measure B and a most preferred one selected for elaboration to level $j+1$. Upon reaching a terminal node or one for which it is known that all paths passing through it are dominated or nonfeasible, the process backs up as usual to the lowest node for which all branches have not been considered. We will consider principally this latter type wherein search can be directed first to the discovery of a feasible solution and then to better and better feasible solutions, although the discussions will generally be equally applicable to the other basic strategies and to mixtures thereof as well.

At this point we can now summarize the approaches of Lawler and Gilmore in the following way. Both approaches employ a search strategy wherein the jth level in the tree corresponds to the assignment of some plant to the jth location, as suggested by the tree of Figure 2. The ordering of locations $j_1, j_2, \ldots, j_n$ is arbitrary or, perhaps as suggested by Gilmore, in accord with some heuristic ordering rule such as by decreasing sums $\sum_{g \neq j} (d_{jg} + d_{gj})$. Given this ordering, both employ the data-dependent

---

*While perhaps making it possible to eliminate from explicit investigation particular subsets of potential solutions in a given problem, feasibility and/or dominance considerations do not change the order of consideration.

†Bookkeeping for the portions of the tree investigated so far would be considerably more extensive, for instance, for a level-by-level type search pattern such as with dynamic programming where all nodes at level j of a tree are elaborated before proceeding to level $j+1$ (or $j-1$), etc.

level-by-level search strategy wherein at level r the node chosen for elaboration to the next level is one for which the bound B_{r+1} is lowest among those not yet elaborated. Both approaches explicitly elaborate the $(n-r)$ nodes branching from the node, evaluate B_{r+1} for each, and repeat the process. Should $B_L > Z_u$ for all nodes at any level L the process backtracks to the lowest level in the tree for which there exists an unelaborated node which is not dominated, and resumes. The difference in the algorithms lies in the bounds B_r used. Lawler being concerned with the general quadratic assignment problem with objective function (1) solves a linear assignment problem to get each $\hat{a}_{ij}^r$ in (7), and then a single linear assignment problem for $[a_{ij}^r]$ to get B_r, as given in (8); when his problem specializes to the single-commodity Koopmans-Beckmann problem, he proceeds in the same manner except that he gets each $\hat{a}_{ij}^r$ directly by ordering elements as described earlier. Gilmore focuses on the single-commodity Koopmans-Beckmann problem and develops B_r either in the manner described for Lawler or in accordance with (9).

f_{ij}

	A	B	C	D
A	—	5	7	2
B	6	—	3	5
C	7	5	—	1
D	2	6	1	—

d_{kl}

	1	2	3	4
1	—	10	20	5
2	18	—	9	4
3	5	6	—	8
4	8	0	15	—

$d_{kl}+d_{lk}$

	1	2	3	4
1	—	28	25	13
2	28	—	15	4
3	25	15	—	23
4	13	4	23	—

ALL $s_{ij} = 0$

Figure 3. Flow and distance data for illustrative problem of Gavett and Plyter.

As an illustration, we will solve the Koopmans-Beckmann problem of Gavett and Plyter [8] shown in Figure 3, computing bounds according to (8). In our solution to this problem we will examine the locations in the sequence A, B, C, D, so that at the first level for instance, we investigate A-1, A-2, A-3, and A-4, and so on. We begin, however, at the 0 level with no assignments. Therefore for

$$A\text{-}1: I = \emptyset, \quad \sum_{(i,k) \in I} S_{\ldots} = 0$$

$$a_{ij}^0 = S_{ij} + \sum_{(k,l) \in I} S_{\ldots} + 1/2\hat{a}_{ij}^0$$

$$= 1/2\,\hat{a}_{ij}^0$$

$$\hat{a}_{A,1}^0 = (7)(13) + (6)(25) + (2)(28) = 297.$$

Using this procedure, we establish $\ldots$ and as a result all

	1	2	3	4
A	297	174	293	152
B	368	254	353	217
C	244	131	245	116
D	156	82	161	73

bounds determined in the following calculations will be exactly twice as large as the true bound. Solving the linear assignment problem defined by this matrix. gives us the resulting matrix with total reduction of 792. The lower bound on the problem is therefore 792/2=396.

	1	2	3	4
A	14	0	5	0
B	20	15	0	0
C	4	0	0	7
D	0	35	0	48

Reduction [illegible]

A-1

$A1: 2a^1_{ij} = 2S_{ijij} + 2\sum_{k\epsilon l} S_{ijk\ell(k)} + \bar{a}^1_{ij}$

for cell B-2 we have.

$$S_{ijij} = 0$$

$$\sum_{k\epsilon l} S_{ijk\ell(k)} = (f_{1,2})(d_{4,n}) = (28)(6) = 168$$

$$\sum_{k\epsilon l} S_{ijk\ell(k)} =$$

	2	3	4
B	168	150	78
C	196	175	91
D	56	50	26

This term represents the interaction of the previous assignment (A-1) with the possible new assignment (B-2)

$$\bar{a}^1_{ij} = (d_{B,C})(f_{2,3}) + (d_{B,D})(f_{2,4})$$

$$= (5)(15) + (6)(4) = 99$$

where $d_{B,C} \geq d_{B,D}$ and $f(2,4) \leq f(2,3)$

$\sum \bar{a}^1_{ij} =$

	2	3	4
B	99	205	139
C	35	98	43
D	39	113	47

and we now have

$2a^1_{ij} =$

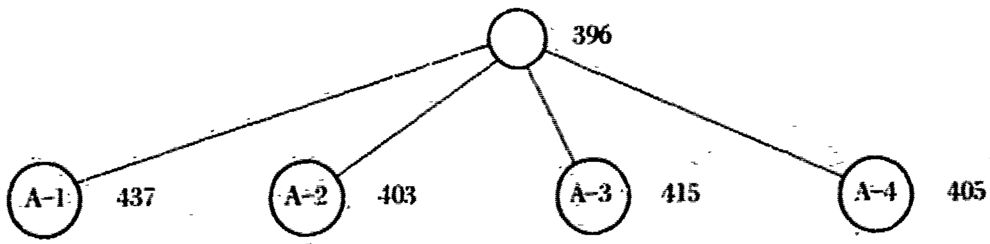
	2	3	4
B	435	505	295
C	427	448	225
D	151	213	99

Solving this linear assignment problem we obtain a reduction of 873 with the resulting matrix.

	2	3	4
B	0	8	0
C	62	21	0
D	0	0	88

To get the desired bound we now divide by two (since the elements in the matrix were $2a^1_{ij}$) and round the resulting fraction up since only integer solutions to the problem are feasible. Similarly we develop level 1 of the enumeration tree.

ALL SOLUTIONS



We select the smallest of these, A–2 to develop first at level two. We now develop cell C–3 given previous selections of A–2, B–1

$$\sum_{k \epsilon I} S_{ijkl(k)} = (f_{3,2})(d_{C,A}) + (f_{3,1})(d_{CB}) + (f_{1,2}) d_{(A,B)}$$

$$\sum_{k \epsilon I} S_{ijkl(k)} =$$

	3	4
C	338	369
D	288	194

$$\Sigma \bar{a}_{ij}^{*} = (d_{CD})(\div 34) = 23$$

$$\Sigma \bar{a}_{ij}^{*} =$$

	3	4
C	23	23
D	23	23

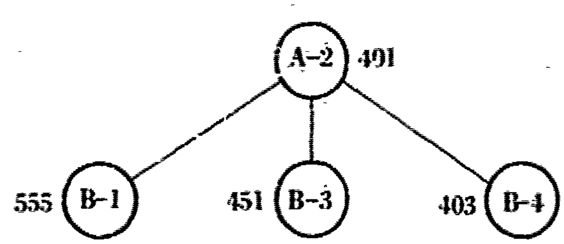
$$2\bar{a}_{ij}^{*} =$$

	3	4
C	699	761
D	599	411

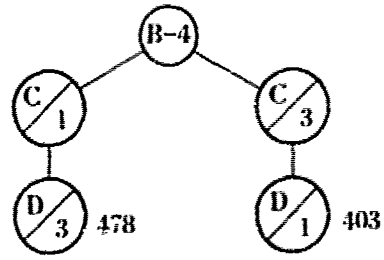
	3	4
C	0	62
D	138	0

$$\text{Reduction } \frac{1110}{2} = 555.$$

The same set of operations for A–2, B–3 and A–2, B–4



The information for level 3, and 4 are obtained by complete enumeration which we designate with the symbol ⊘. We begin with B–4.



The value of 806 for A-2, B-4, C-3, D-1, is shown to be the optimal solution (Figure 4).

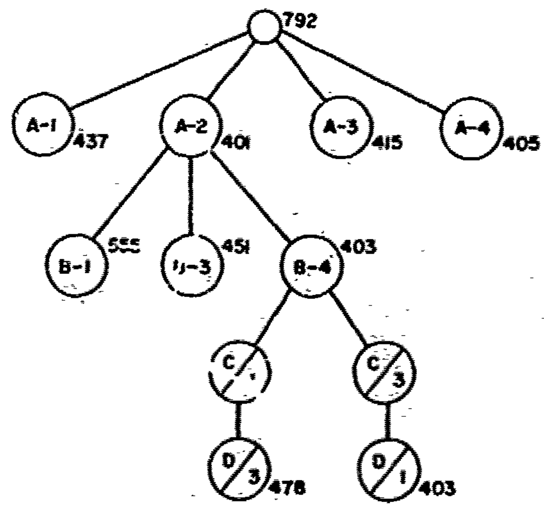


FIGURE 4. Tree elaborated for problem of Figure 3 using Gilmore-Lawler algorithm with bounds of Equation (8).

In contrast, by using the less stringent (but more easily evaluated) bounds of Gilmore (9) the result is as shown by the tree of Figure 5. As illustrations, we will now demonstrate the calculations for a

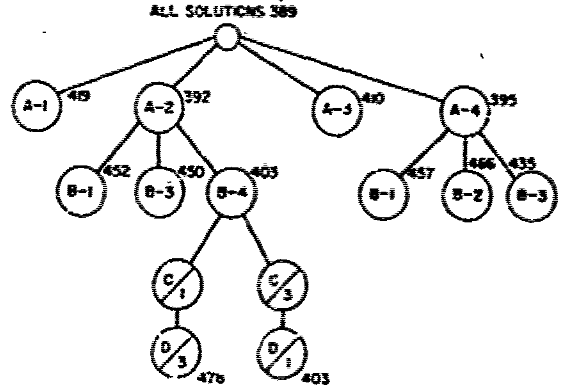


FIGURE 5. Tree elaborated for problem of Figure 3 using Gilmore-Lawler algorithm with bounds of Equation (9)

bound on (i) all solutions, (ii) solutions with A–4, and (iii) solutions with A–4, B–3 are as follows:

$$\text{(i)}\quad 7\cdot 4+6\cdot 13+6\cdot 15+5\cdot 23+2\cdot 25+1\cdot 28=389$$

$$\text{(ii)}\quad 7\cdot 4+6\cdot 13+2\cdot 23+6\cdot 15+5\cdot 25+1\cdot 28=395$$

$$\text{(iii)}\quad 6\cdot 23+7\cdot 4+2\cdot 13+6\cdot 15+5\cdot 25+1\cdot 28=435.$$

The resulting tree is seen to have a greater number of nodes than the former, but since the evaluation of each is less time-consuming the total problem-solving time could be smaller.

III. EXTENSIONS OF THE SINGLE-ASSIGNMENT ALGORITHM

Turning to prospective improvements in the problem-solving procedures which have been discussed, let us review the steps in the Gilmore-Lawler algorithm at a node on level $(v-1)$ in the tree. For each of the $(n-v+1)$ assignments (i, j) that can be made a lower bound B_v is determined according to (8). To determine each of the values B_v requires the formation of an $(n-v+1)x\ (n-v+1)$ matrix $\mathbf{A}^v$ and the solution of the linear assignment problem which it defines. To get each of the elements a_{ij}^v requires in turn the solution of an $(n-v)$ dimensional assignment problem (which in the Koopmans-Beckmann problem can be accomplished by simply sequencing the relevant flow and distance values and forming the inner product.) To make an assignment at this node thus entails the solution of $(n-v+1)$ assignment problems of dimension $(n-v)$ and $(n-v+1)(n-v)^2$ problems of dimension $(n-v-1)$. By expending less computation effort in making an assignment at each stage it may, however, be possible to achieve overall improvement in problem solving. In the following discussion we shall continue to employ the same level by level search strategy, choosing at each level a node with a lowest bound, but shall consider alternate ways of assessing the lower bounds.

In (8) the value Z_v^* for an optimal assignment solution to the problem defined by matrix $\mathbf{A}_v$ was employed in developing bound B_v, but in general any value Z_v constituting a lower bound on Z_v^* may also be used. One such bound less stringent than Z_v^* can be computed with little effort by the matrix reduction method used by Little, et al. [20]. This method rests on the fact that if $T(g)$ is the cost of an assignment with respect to a matrix A and if $T'(g)$ is the cost of that assignment with respect to matrix A' which is formed by subtracting the constant b from each element of one row or column, then $T(g)=T'(g)+b$, and the optimal assignments under both matrices are the same. By subtracting appropriate constants from each row and column, a matrix A'' of nonnegative elements with at least one zero in each row and column can be obtained.* Such a matrix they have termed a "reduced matrix" and the sum of the constraints subtracted in forming the matrix, the "amount of reduction." If $T''(g)$ is the total cost of an assignment with respect to the reduced matrix A'' and R is the amount of reduction incurred in reducing A, then $T(g)=T''(g)+R$. Since all elements in A'' are nonnegative, $T''(g)\geq 0$ for all assignments g, and therefore the amount of reduction R constitutes a lower bound on the optimal value of the assignment problem defined by A.

We will denote by A_v'' a reduced matrix for A_v and by Z_v'' the reduction achieved in reducing it. Z_v'' may be used in place of Z_v in determining B_v.

---

*This can be accomplished, for instance, by first subtracting from each column the smallest element in the column and then subtracting from each resulting row the smallest element in the row. In general, however, the reduced matrix and the amount of reduction are not unique, but may be dependent on the order in which rows and column are reduced

Moreover, between Z_r^* and Z_r^* there are a number of values Z_r^t which may be used. Of special interest are those derived during solution of the linear assignment problem by a dual algorithm since, for successive iterations $t, t+1, \ldots,$ of the algorithm, the value of the objective function Z_r^t is non-decreasing: $Z_r^t \leq Z_r^{t+1} \leq \ldots \leq Z_r^*$. With such an algorithm problem-solving can terminate should the condition

$$Z_r^t + \sum_{i,k\in I} S_{i\ell(i)k\ell(k)} \geq Z_b,$$

become satisfied for any t, since all paths passing through the node associated with A_r must then be dominated. Algorithms of this type include, for example, the Hungarian method [4], the network flow algorithm of Ford and Fulkerson [6] and the flow algorithm as improved by Sprague [29]. At each iteration in these dual algorithms Z_r^t is the amount of reduction associated with a matrix of nonnegative coefficients A_r^t derived from the original, a matrix in which $a_{ij}^t = 0$ for each $x_{ij} = 1$ in the optimal solution.

Besides the choice of the amount of reduction to perform on a matrix $A_r = \|a_{ij}^r\|$ there are numerous alternatives for selecting the elements a_{ij}^r to be used in assessing B_r. As was noted earlier, in general any value a_{ij}^r which results from use of an appropriate lower bound in (7) for a_{ij}^r is permissible. Thus, for example, in cases where in developing $A_r = \|a_{ij}^r\|$ it is not possible to determine a_{ij}^r simply by by sequencing the flow and distance elements and forming their inner product, it may prove efficient to determine lower bounds on the a_{ij}^r in this same way, and then proceed to solve the resulting matrix A_r as discussed. Another possibility is to simply set $A_r = \|a_{ij}^{r-1}\|$, $i \neq k$ and $j \neq q$ where (k, q) is the assignment made in passing from level $r-1$ to level r, and then to employ the bound:

$$B^r = Z_r^t + a_{kq}^{r-1} + \sum_{i,k\in I} S_{i\ell(i)k\ell(k)},$$

where Z_r^t is a lower bound on the problem defined by A_r and I is the set of assignments existing when the elements a_{ij}^{r-1} were determined. Or, in general,

$$B_s^r = Z_r^t + \sum_{i=s}^{r-1} a^r_{j_i\ell(j_i)} + \sum_{i,k\in I} S_{i\ell(i)k\ell(k)}, \tag{10}$$

where* $A_{r,s} = \|a_{ij}^s\|$, and $a^s_{j_i\ell(j_i)}$ are the coefficients of the assignments $(j_s, \ell(j_s)), (j_{s+1}, \ell(j_{s+1})), \ldots, (j_{r-1}, \ell(j_{r-1}))$ which have been made and Z_r^t is a lower bound in the problem defined by $A_{r,s}$. And between these extreme alternatives of determining a minimum value for every a_{ij}^r according to (7) at level r and of simply using a_{ij}^s from a previous stage there is, for example, the alternative of re-computing only selected a_{ij} perceived to be critical,† together with others.

---

*The bound in (10) follows directly from the fact that the sum of the coefficients a_{ij} in A_r for any feasible linear assignment solution constitutes a valid lower bound on the cost of that assignment in the quadratic problem.

†As the potential variability in a cost coefficient a_{ij} diminishes with successive assignments the potential importance of updating its value may also diminish. For example, referring to (7) it is seen that the only variability in the coefficient a_{ij} from level to level derives from the product $\sum_{p\in I} f_{ip} d_{jp'(p)}$, which decreases with increasing r. By letting

$$f_i^* = \min_j \{f_{ij}\},\ d_i^* = \min_k \{d_{jk}\},\ f_{ij} = f_i^* + f'_{ij},\ \text{and}\ d_{ij} = d_i^* + d'_{ij},$$

the sum becomes

$$\sum_{p=1}^{n} f_{ip} d_{jp'(p)} = \sum_{p=1}^{n} (f_i^* + f'_{ip})(d_j^* + d'_{jp'(p)})$$

$$= \text{Constant} + \sum_{p} f'_{ip} d'_{jp'(p)}.$$

The maximum variability is thus $\sum_{s} (f_{i(s)} - f_{i(s+1)})(d_{j(s)} - d_{j(s+1)})$, which can perhaps be used to assess the potential importance of updating the coefficient a_{ij}.

Continuing further the discussion of alternate means of bounding, recall that the situation we have been discussing was that in which we had arrived at a node at level $(r-1)$ and, in the manner of Lawler and Gilmore, were making each of possible $(n-r+1)$ assignments (j, i) at level r and evaluating through means of an appropriate matrix A_r a bound B_r for each of the $(n-r+1)$ nodes. Any of the ways for getting the elements for the matrices A_r and any degree of reduction could be employed in each case. However, while perhaps resulting in less stringent bounds, a value for each B_r at level r can be assessed at level $(r-1)$ without first generating each of the matrices A_r, hence reducing the computational effort preparatory to making a next assignment. For if A_{r-1} is an appropriate assignment matrix for the problem at level $(r-1)$ and $A^l_{r-1} = \|a^{l,r-1}_{ij}\|$ is any matrix with nonnegative elements derived from it through one or more stages of reduction, then a lower bound on solutions passing through the node at level r which results from the assignment (i, j) is

$$B^l_{r-1}(i, j) = a^{l,r-1}_{ij} + Z^l_{r-1} + \sum_{i,k\epsilon I} S_{\ldots} \tag{11}$$

where Z^l_{r-1} is the amount of reduction incurred in reducing matrix A_{r-1} to A^l_{r-1}. In practice A^l_{r-1} would most likely be the reduced matrix which results from simply reducing rows and columns, or the matrix A associated with an optimal assignment solution in which $a_{ij}=0$ for all $x_{ij}=1$ in the optimal solution. To facilitate discussion, we will assume at least the former so that there exists at least one zero element in each row and column of A^l_{r-1} although this in no way limits the generality of the discussion.

To employ in this framework the search strategy of Gilmore and Lawler which selects an assignment (i, r) which has a lower bound, we simply choose an assignment corresponding to a zero element in the column of A_{r-1} representing location r. We then proceed to formulate a matrix A_r for this one node, generating some or all of the remaining $(n-r)$ matrices at this level at a later point in the search process only if not dominated.

To generalize somewhat beyond the search strategy of Gilmore and Lawler and make search more dependent on the data, we may select from all candidate assignments (i, j) at level $(r-1)$ a next assignment, not limiting choice to location r. Assuming A_{r-1} is a reduced matrix, however, there are at least $(n-r-1)$ zero elements, at least one for each row and column. Therefore, additional criteria are required for choosing among the zero elements.

A very effective criterion is that of Little et al. [20] which employs what is termed an alternate cost. At each point throughout the search process the selection of an assignment (i, j) partitions the set of all potential solutions into two subsets, one of all potential solutions which includes (i, j) and the other of all potential solutions which does not. At level $r-1$ a lower bound on the cost of potential solutions in the first subset is

$$Z^l_{r-1} + \sum_{p,k\epsilon I} S_{\ldots}$$

since (i, j) is a zero element. On the other hand, since one element must eventually be selected from each row and each column of the assignment matrix, a lower bound for potential solutions in the second subset is

$$E^l_{r-1}(i, j) = Z^l_{r-1} + \sum_{p,k\epsilon I} S_{\ldots} + \min_{x\neq i}\{a^l_{(x),(j)}\} + \min_{y\neq j}\{a^l_{(i),(y)}\}. \tag{12}$$

We will refer to the quantity $E^l_r(i, j)$, a lower bound on the objective function for all potential alterna-

tives to the pair (i, j), as simply the alternate cost for the pair (i, j). According to the criterion of Little, et al. the zero element is chosen for which the alternate cost is the greatest. Thus, search proceeds stage by stage selecting elements according to their cost criterion until either a terminal node is reached or one for which it is known that all potential solutions passing through it are dominated. At this point the search process backtracks to the first node for which the alternate cost is less than the total completion time of the best feasible sequence discovered so far, sets $a_{kp} = M$ for the assignment just investigated, and then resumes.

As a computational consideration it is noted that if $E'_{r-1}(i, j) \geq Z_s$ then the assignment (i, j) must necessarily be included in every nondominated path passing through the node. If this is true for two or more assignments, then it is unnecessary to explicitly consider nodes for each of these, but rather make all such assignments immediately (jumping levels in the tree) and then proceed to establish matrix A_r for the remaining choices.* In the special event that this is true for all zero elements in an optimal assignment solution at a given node, then the only further consideration that need by given the node is to evaluate the quadratic assignment solution defined by this assignment.

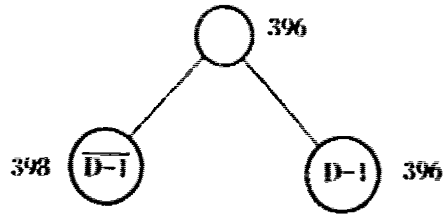
In summary, there are many bounding alternatives that may be employed within the tree search algorithm. Basically, as we have seen, these concern alternative ways for determining the elements in matrix A_r at a given node; alternative degrees of reduction to be applied to it; and choices regarding the dominance tests to be made on the basis of the resulting bounds. Further, as has been discussed, there are a number of alternative search strategies ranging from fixed, data-independent strategies to the level-by-level strategy with pre-specified levels of Gilmore and Lawler, and to the general level-by-level strategy with variable levels. In addition, there is the alternative of stopping to evaluate the quadratic cost of a feasible solution which results whenever a feasible solution to the linear assignment problem is determined at any node; for if $Z_a < Z_s$ a better feasible solution has been discovered and the lower value of Z may be used to make potentially more stringent the dominance tests in reducing subsequent search.

To illustrate these extensions we again solve the problem of Figure 3. In the algorithm to be used the search strategy is the general level-by-level strategy except that when we reach level $(n-2)$ we shift to a data-independent strategy of exhaustively enumerating all feasible assignments. Beginning at level 0 and at every level thereafter, we establish matrix A_r by determining optimal values of the elements a_{ij} and then reducing fully to an optimal assignment solution.

We then solve a second time the same problem, illustrating the possibility of evaluating the feasible quadratic assignment solution defined by the optimal linear assignment. In this procedure we also review the alternate costs at every node to indentify variables x_{ij} which must necessarily have value $x_{ij} = 1$. For all such variables a level in the tree is jumped.

We begin with the matrix developed previously and the lower bound on all solutions of $\frac{792}{2} = 396$. An examination of the matrix shows that cells A–2, A–4, B–3, B–4, C–2, C–3, D–1, and D–3 may be selected at zero incremental cost. The alternate cost of D–1, that is 4, is higher than any other so we make the first branch at this cell with alternate cost $\frac{792+4}{2} = 398$.

*Note that before proceeding to establish the new matrix it might prove worthwhile to re-evaluate the alternate costs of the remaining zero elements and re-check whether or not $E'_{r-1}(i, j) \geq Z_s$ for any additional zero element.



We then calculate a_{ij}^{1} as before. The elements are

$\sum_{k,l} S_{ik\,jl} =$

	2	3	4
A	56	50	26
B	168	150	78
C	28	25	13

$\Sigma \bar{a}_{ij}^{1} =$

	2	3	4
A	118	243	166
B	99	205	139
C	103	220	143

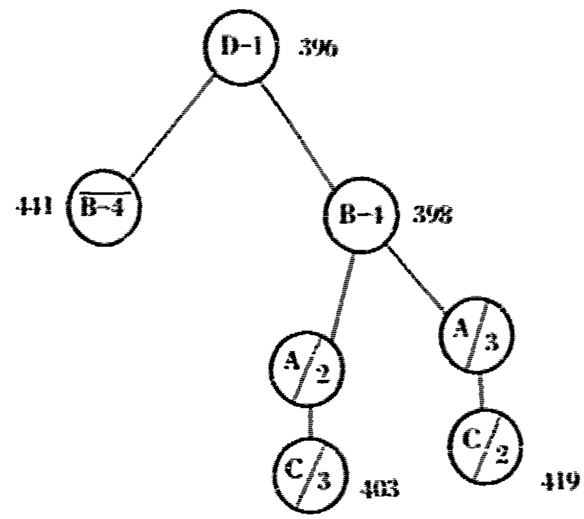
$2a_{ij}^{1} =$

	2	3	4
A	230	343	218
B	435	505	295
C	159	270	169

Solving the linear assignment problem associated with this matrix we obtain the following solution with a total reduction of $\frac{795}{2} = 398$.

	2	3	4
A	0	2	0
B	128	87	0
C	0	0	22

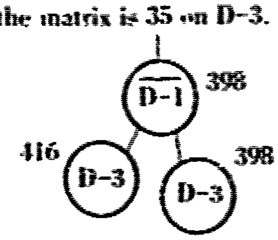
The assignment with the largest alternate cost is B–4 with 87. We therefore branch on B–4, then enumerate all possible alternatives following D–1, B–4.



At this point all completions are bounded by the solution D–1, B–4, A–2, and C–3, with the exception of $\overline{D-1}$. We next modify the initial matrix to reflect the condition $\overline{D-1}$, by adding M (a large number) to the D–1 element. The resulting matrix is solved as a linear assignment problem below, with a reduction of $\frac{796}{2}=398$.

	1	2	3	4
A	4	0	40	0
B	10	15	35	0
C	0	6	41	13
D	M	0	0	13

The highest alternate cost in the matrix is 35 on D–3. We branch as below



The matrix given D–3 is

$2a_{ij}^1=$

	1	2	4
A	359	256	198
B	318	344	365
C	281	198	139.

Simple row and column reduction gives a total reduction of $\frac{822}{2} = 411$ and the following matrix:

	1	2	4
A	20	0	1
B	91	0	80
C	0	0	0.

This is adequate to show that all solutions containing D–3 are bounded, and that we have discovered the optimal solution. We have now completely elaborated the enumeration tree as shown in Figure 6.

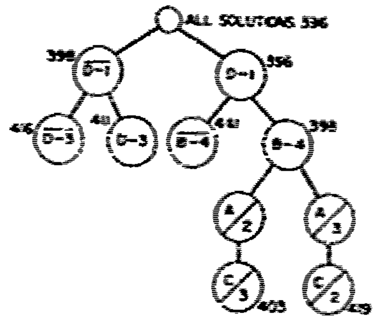
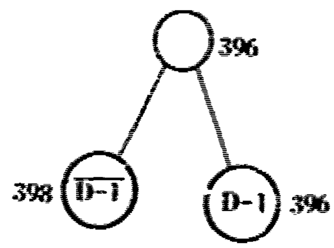


FIGURE 6. Tree elaborated for problem of Figure 3 with alternative single-assignment algorithm.

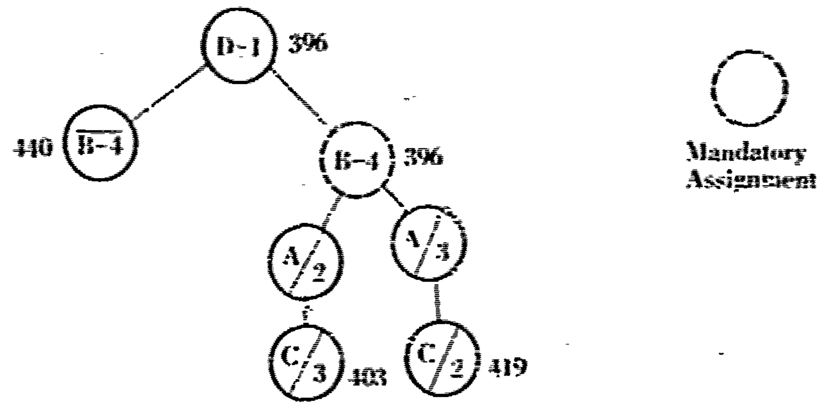
As noted earlier if $E_t^s(i, j)$ in (12) is greater than or equal to Z_s then the assignment (i, j) must necessarily be included in every nondominated path passing through the node. We will now modify the calculations in our sample problem to include this test. We begin again with the initial assignment solution D–1, C–2, B–3, A–4, with a total reduction of $\frac{792}{2} = 396$ and an actual cost of $\frac{890}{2} = 445$. At this point we set $Z_s = 445$. The first branch is D–1 as before and we form and reduce the second matrix and solve for an optimal linear assignment.



	2	3	4
A	0	2	0
B	128	87	0
C	0	0	22

Reduction = 796

The exact cost of the current assignment solution, D–1, A–2, B–4, C–3 is 403 and this becomes the new value for Z_0. An evaluation of the alternate costs shows that B–4, with a cost of 87 must be in an optimal solution if D–1 is. We then enumerate the remaining two assignments.



An evaluation of the $\overline{D-1}$ branch in the original reduced matrix as shown below

	1	2	3	4
A	14	0	5	0
B	20	15	0	0
C	4	0	0	7
D	M	35	0	48

now indicates that the alternate cost of D–3 is 35 and since $\frac{792+25}{2} > Z_0 = 403$ then D–3 must be in any solution which is superior to Z_0. If the alternate cost of B–4 is updated we find it is now $\left(\frac{792+15}{2}\right) > 403$ and therefore B–4 must be in any superior solution. By the same argument A–2 and C–1 are also assigned. We now can complete the tree based solely on information in the first matrix created and on an evaluation of the complete solution D–3, B–4, A–2, C–1. The resulting enumeration tree is shown in Figure 7.

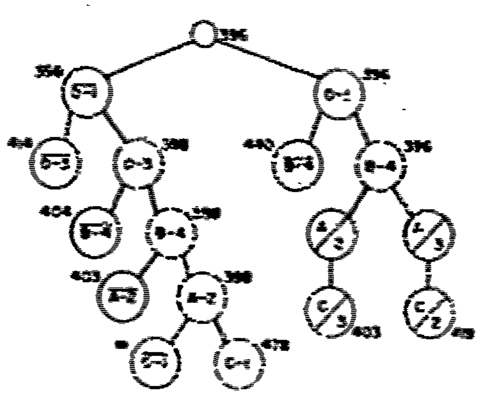


FIGURE 7. Tree elaborated for alternative single assignment algorithm together with testing for mandatory assignments.

We will now show an alternate method for bounding the D-1 path which involves the updating of particular cells in the cost matrix. To begin we solve the assignment problem for the best solution not including D-1, getting the following matrix:

	1	2	3	4	
A	4	0	40	0	
B	10	15	35	0	
C	0	6	41	13	
D	M	0	0	13	Reduction $\frac{796}{2} = 398.$

Since D-3 has the highest alternate cost we select that assignment and then update only the other elements of the assignment solution, C-1, A-2, B-4, on the assumption that D-3 is selected.

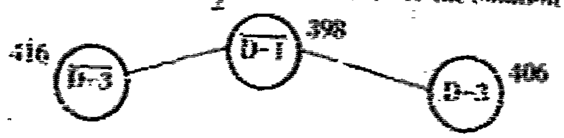
This results in the following matrix:

	1	2	3	4	
A	297	256	293	152	
B	368	254	353	365	
C	281	131	245	116	
D	M	82	161	73	UPDATED

with assignment solution

	1	2	3	4
A	46	89	47	0
B	0	79	0	94
C	65	0	35	0
D	M	3	0	6

and total reduction of $\frac{812}{2} = 406$. Therefore the solution with D-3 is bounded.



For this particular problem this method resulting in the solution of Figure 7 is the most efficient of the methods discussed in this section. Although the tree is larger than that of Figure 6 it is generated completely from the first matrix and from two completely enumerated solutions.

In concluding discussion of single-assignment algorithms it should be emphasized that while we have been illustrating ideas by means of a simple Koopmans-Beckmann problem (5) with all costs $c_{ij} = 0$, the ideas and algorithms are perfectly general and apply equally as well to the more generalized Koopmans-Beckmann problem with cost function (6) as well as to the general quadratic assignment problem with cost function (1).

As remarked in the Introduction, there are sometimes constraints to be satisfied in addition to those of (2)–(4). To the extent that the conditions of these constraints can be completely stated by setting $s_{ijkq} = M$ for appropriate i, j, k and q, the algorithms as discussed can be utilized without change. For algorithms which, for problems where $s_{ijkq} = f_{ik}d_{jq}$, determine elements a_{ij}^{kq} simply by appropriately sequencing the elements f_{ik} and d_{jq} there are now at least three options: First, continue to determine a_{ij}^{kq} in the same way, the result being a lower bound on the true minimum value; second, determine the true minimum by solving an assignment problem with $s_{ijkq} = M$ where appropriate; or third, determine the sequencing of f_{ik} and d_{jq} in the present way and make adjustments for inadmissable pairings $f_{ik}d_{jq}$ which result. In each case when $s_{ijkq} = M$ we will set $a_{ij}^{kq} = M$ whenever assignment (kq) is made, and $a_{kq}^{ij} = M$ whenever (ij) is made. When constraints involving three or more assignments are present, we can proceed in the same way except that the constraints become explicitly represented in the problem through the s_{ijkq} and a_{ij}^{kq} only when a sufficient number of assignments have already been made to enable identification of these nonfeasible assignments.

IV PAIR-ASSIGNMENT ALGORITHMS

In contrast to the algorithms which have been discussed in the previous sections, both Land [16] and Gavett and Plyter [8] have developed algorithms in which search proceeds on the basis of a controlled enumeration of the variables $y_{ijkq} = x_{ij} \cdot x_{kq}$ where as before each variable x_{ij} denotes the locating of plant i at location j. These authors too view the underlying problem as a linear assignment problem, but one of assigning a *pair* of plants i and k to locations j and q. In both instances [8, 16] the algorithms developed apply to the symmetric Koopmans-Beckmann problem with $S_{ijkq} = S_{iqkj} = f_{ik}d_{jq}$.

In Figure 8 is shown the relevant assignment matrix for the problem of Figure 3. In general, there are $n(n-1)/2$ pairs of plants and pairs of locations in the problem.

LOCATION / PAIR	2–1 / 1–2	3–1 / 1–3	4–1 / 1–4	3–2 / 2–3	4–2 / 2–4	4–3 / 3–4
PLANT PAIR						
A–B	168	150	78	90	24	138
A–C	196	175	91	105	28	161
A–D	56	50	26	30	8	46
B–C	140	125	65	75	20	115
B–D	168	150	78	90	24	138
C–D	28	25	13	15	4	23

FIGURE 8. Data for problem of Gavett and Plyter represented in terms of pairs of assignment.

However, there are many feasible solutions to this assignment problem which are not feasible solutions to the original quadratic assignment problem. For example it is entirely acceptable in the linear assignment problem for plants A and B to be assigned locations 1 and 2 and plants A and C locations 3 and 4, a solution clearly infeasible for the original quadratic problem. For a feasible solution to the original

problem we must therefore, affix to the linear assignment problem the following additional constraints:

If $$y_{ijkq} = 1$$

Then

(13)
$$\begin{aligned} y_{ivp\ell} &= 0 & y_{vjp\ell} &= 0 \\ y_{ujp\ell} &= 0 & y_{iup\ell} &= 0 \\ y_{uvk\ell} &= 0 & y_{uv\ell k} &= 0 \\ y_{uvpq} &= 0 & y_{uvqp} &= 0, \end{aligned}$$

where $i \neq u \neq j, \quad i \neq v \neq j, \quad k \neq p \neq q$ and $k \neq \ell \neq q$.

Operationally both the algorithm of Land and that of Gavett and Plyter commence by determining an optimal linear assignment solution for the matrix A_0, and determining a reduced matrix A_0'' with nonnegative entries in which $a''_{ijkq} = 0$ for all variables $y_{ijkq} = 1$ in the optimal solution. Thereafter Gavett and Plyter employ only a row and column-reduced matrix A_r at each node, and Land employs only a column-reduced matrix at each node. As in the procedures discussed in the previous sections, both of their algorithms proceed level by level in the tree, committing one new pair (i.e., setting $y_{ijkq} = 1$) to the solution at each level, and backtracking to the lowest level in the tree having an unevaluated branch. In selecting the pair to be committed at a given level in the tree Gavett and Plyter use the alternate cost method of Little et al. [20], while Land [16] always selects from the column having the fewest number of feasible elements in the column-reduced matrix A_r a zero element having the largest alternate cost (based only on alternate costs in the same column). After committing a pair to the solution at a given node in the tree (i.e., setting $y_{ijkq} = 1$) feasibility condition (13) is invoked by setting the cost $c_{efgh} = M$ for all $y_{efgh} = 0$ specified in (13), and the resulting matrix used as matrix A_{r+1} at the next level. In a variation of the search procedures discussed heretofore, however, Gavett and Plyter, after selecting the assignment pair and hence the variable y_{ijkq} at node v, apply (13) for the branch $y_{ijkq} = 1$ and reduce the resulting matrix A_{r+1} to get a lower bound on the cost of solutions with $y_{ijkq} = 1$; if the resulting bound exceeds the alternate cost of the assignment ($ijkq$) they will at this point in the search pursue the branch for $y_{ijkq} = 0$ rather than that* for $y_{ijkq} = 1$.

This latter point is readily illustrated, for instance, in Figure 9 which shows the tree elaborated

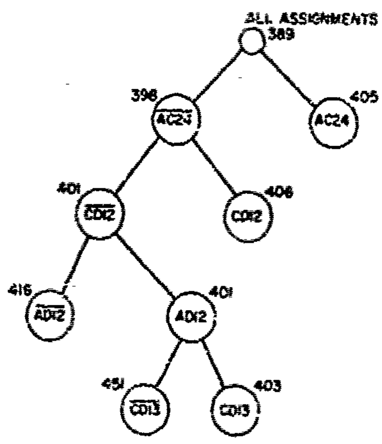


FIGURE 9. Tree elaborated for problem by Gavett and Plyter algorithm.

*In comparison with the other search strategies that have been discussed, this strategy may entail the elaboration of a longer path in the decision tree and require longer problem-solving time to determine a first feasible solution.

by the Gavett and Plyter algorithm for the example of Figure 3. Since the detailed calculations underlying the development of this tree are contained in [8] we shall omit them here.

As in the case of single-assignment algorithms there are a number of alternatives to these pair-assignment algorithms which may result in more efficient algorithms. For example, all of the alternatives discussed earlier concerning the extent to which a linear assignment matrix is reduced at a node in the tree are applicable in the present problem. Similarly the use of alternate costs to identify mandatory assignments and the jumping of levels in the tree on the basis thereof, is equally appropriate in the present case. Of course in the present problem the alternatives for determining the costs $a^{[r]}_{ijkq}$ are inapplicable since $A_r = a^{[r-1]}_{ijkq}$ for all v, i.e. the value $a^{[r]}_{ijkq}$ is the actual cost of assigning plant i to location j and plant k to location q rather than simply a lower bound, and hence need not be updated.

In concluding discussion of this class of algorithms we comment on their extension to the nonsymmetric quadratic assignment problem in which $s_{ijkq} \neq s_{iqkj}$. For this problem we have the associated linear problem:

Minimize $$\sum_{(ik),(jq)} (s_{ijkq} y_{ijkq} + s_{iqkj} y_{iqkj})$$

Subject to:

(14) $$\sum_{(jq)}^{n(n-1)/2} (y_{ijkq} + y_{iqjk}) = 1 \qquad \text{all } (ik)$$

$$\sum_{(ik)}^{n(n-1)/2} y_{ijkq} + \sum_{(ik)}^{n(n-1)/2} y_{iqjk} = 1 \qquad \text{all } (jq)$$

and $$y_{ijkq}, y_{iqjk} = 0, 1 \qquad \text{for all } i, j, k, q.$$

upon which are imposed, as before, constraints (13).

As discussed in more detail for a related, multi-facility production requiring problem (or multi-salesman traveling-salesman problem) in [25], when formulated as a linear programming problem (14) has a total of $n(n-1)$ variables and activity vectors, the vectors for each pair y_{ijkq} and y_{iqkj} being identical except for their cost. Being linearly dependent, at most one of these vectors in each pair can appear in an optimal feasible solution to (14), this necessarily being the vector in the pair with the smaller cost. Therefore setting $\bar{s}_{ijkq} = \min\ (s_{ijkq}, s_{iqkj})$ it follows that an optimal solution to (14) can be obtained by solving the $n(n-1)/2 \times n(n-1)/2$ linear assignment problem with costs $\|\bar{s}_{ijkq}\|$.

Operationally, then, problem-solving for the nonsymmetric problem can proceed as for the symmetric except that at each node in the process the cost matrix to be used is composed of the presently minimum elements, $\|\min\ (s_{ijkq}, s_{iqkj})\|$. To this matrix can be applied any degree of reduction as in the symmetric problem. Upon selecting a pair of assignments to commit to the quadratic problem solution, *all* costs (both for a variable y_{ijkq} and its interchange y_{iqkj}) are updated as required to reflect the feasibility conditions in (13); for any pair therefore, $\min\ (s_{ijkq}, s_{iqkj})$ may increase for the next node. Otherwise the only difference between the symmetric and the nonsymmetric problems concerns the alternate cost of an assignment: in the nonsymmetric case a valid alternative to the assignment y_{ijkq} may be the interchanged assignment y_{iqkj} so that in this case the alternate cost is the minimum of that as evaluated for the symmetric problem and the cost of the interchanged assignment.

V. PAIR-EXCLUSION ALGORITHMS

In all of the algorithms discussed up to this point, problem-solving has proceeded on the basis of a stage by stage commitment of assignments to the solution of the problem, each such assignment representing a level in the decision tree. Upon backtracking a particular assignment would then be excluded from the solution and the forward, assignment process resumed. This has been the nature of the process for both the single-assignment and the pair-assignment algorithms. In this section we conclude the paper with an algorithm in which problem-solving proceeds on the basis of a stage-by-stage exclusion of assignments from a solution to the problem.

More specifically, let us consider the quadratic assignment problem as formulated in the previous section. Suppose for this problem an optimal assignment has been determined for the linear assignment portion of the problem. If for this assignment conditions (13) are satisfied for every $y_{ijkq} = 1$ in the solution (i.e., all pairs result in each plant being assigned to one location, and no location having more than one plant assigned to it) then this solution represents an optimal, feasible solution to the original quadratic assignment problem and problem-solving is complete. Otherwise there exists one or more conflicting assignments in this solution rendering infeasible the solution to the quadratic assignment problem.

$$\text{COST} = 78 + 28 + 50 + 115 + 90 + 28 = 389$$

	1–2 2–1	1–3 3–1	1–4 4–1	2–3 3–2	2–4 4–2	3–4 4–3
AB	31	16	0°	0	0	8
AC	55	37	9	11	0°	27
AD	3	0°	32	24	68	0
BC	18	6	2	0	11	0°
BD	31	16	0	0°	0	8
CD	0°	0	44	34	89	2

(a)

	1–2 2–1	1–3 3–1	1–4 4–1	2–3 3–2	2–4 4–2	3–4 4–3
AB	31	16	M	0°	0	8
AC	55	37	9	11	0°	27
AD	3	0°	32	24	68	0
BC	18	6	2	0	11	0°
BD	31	16	0°	0	0	8
CD	0°	0	44	34	89	2

(b)

FIGURE 10. Optimal linear assignment matrices to problem with data in Figure 8; (a) all assignments admissable, (b) (AB, 14) inadmissable.

As an example, Figure 10(a) shows a reduced matrix for the illustrative problem in Figure 8 in which the optimal linear assignment is indicated by the cells with alternate costs represented in the upper right hand corner. As is readily verified, this optimal linear assignment is not feasible for the quadratic problem, e.g., assignment (*AB*, 14) is inconsistent with (*BD*, 23), (*BC*, 34) with (*AD*, 13) and (*CD*, 12). In an optimal feasible quadratic assignment, then, it must be true that at least one of the assignments in this optimal linear assignment will not be present. We can therefore subdivide the total set of feasible quadratic assigements into those that do not include the assignment (*AB*, 14), those that do not include

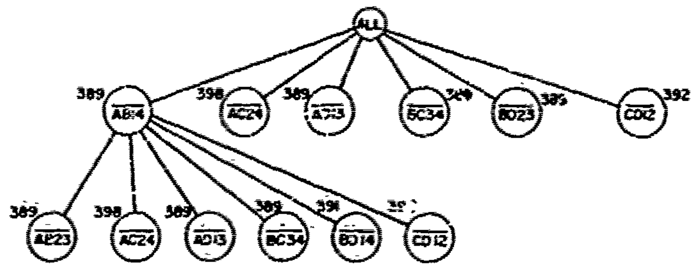


FIGURE 11. Partial tree elaborated by pair-exclusion algorithm.

$(AC, 24)$, and so on, for each of the present assignments. The result, in terms of a tree, is as shown as the first level of nodes in Figure 11. If for each subset we now determine the best feasible quadratic assignment among solutions in that subset, the best among these is an optimal solution to the original problem.

Beside each of the nodes on level 1 in Figure 11 is shown a lower bound on the cost of solutions in the subset equal to the cost of the optimal solution in Figure 10(a) (the amount of the reduction) plus the alternate cost of the particular assignment which, as indicated by the node, is to be excluded. Suppose we now choose for elaboration one of these nodes for which this bound is minimum, say $\overline{AB14}$. Making inadmissable this assignment in the cost matrix in Figure 10(a) (i.e., giving a large cost of M) and solving for an optimal assignment to the resulting problem, there results the matrix in Figure 10(b). Checking the assignment which results, this solution is not feasible for the quadratic problem either; the result is the tree with the new level of nodes as shown in Figure 11.

In a similar manner, we can now proceed to select any of these nodes with lowest bound, solve the assignment problem and check it for feasibility, continuing until a node is reached for which the optimal linear assignment is a feasible quadratic assignment. At this point we would then backtrack and resume with a node whose lower bound was less than the value of the quadratic assignment solution, continuing in this manner until the complete tree has been considered.

In general, it is difficult to anticipate the performance of this type of algorithm relative to the commitment types as discussed in the earlier sections. For the related, basic traveling salesman problem this general approach has proved significantly more efficient than stage-by-stage commitment algorithms [4, 28, 29]. Undoubtedly this is due at least in part to the fact that, in the words of Shapiro [28], the optimal traveling salesman solution is frequently quite "close" to the optimal linear assignment solution in the respect that a large majority of assignments in the former are present in the latter, so that relatively small decision trees need be explicitly elaborated. In addition it is due in part to the existence of an efficient dual algorithm for solving the linear assignment problems at each node [29]. In the present problem this latter element will be equally important, but, on the other hand, it is not apparent that the optimal quadratic assignment will be "close" to the optimal linear assignment, i.e., that in optimal linear assignments the conditions in (13) will commonly be automatically satisfied.

To pursue discussion in greater detail, there are a number of choices which must be made in specifying a particular "pair-exclusion" algorithm. What search strategy is to be employed in selecting a node in the tree to elaborate next? Given the conflicts in an optimal linear assignment solution at a given node, how should solutions be subdivided into subsets for further evaluation? Which branch emanating from a node (i.e., which subset) should be considered first?

For specificity let us assume for discussion purposes that the search strategy used is the same as that which has been assumed in all of the other algorithms discussed. That is, we proceed downward in the tree one level at each successive stage, choosing at each stage for elaboration the node having the

smallest lower bound on the cost of solutions represented by the node. Upon reaching the bottom of the tree or reaching a node for which there exists no feasible, nondominated solutions the process backtracks to the lowest level in the tree for which there is an unevaluated node and resumes. There remains, then, the determination of subsets at nodes and the assessment of lower bounds for the solutions contained in the resulting subsets.

While subdividing the subsets into the n subsets on the basis of the n assignments in the linear solution is perhaps the easiest subdivision to specify at a node (as was done in the illustration) it is by no means the only possibility nor probably the most desirable subdivision. In general, any subdivision into subsets at a node is permissable which excludes at least one conflict present among the present

Assignment	*Alt. cost*	*Conflict*	*Alt. cost*
AB14	389	BD23	389
AB14	389	CD12	392
AC24	398	AD13	389
AC24	398	BD23	389
AD13	389	BC34	389
BC34	389	CD12	392

FIGURE 12. Conflicts between pairs of assignments in optimal linear assignment solutions to illustrate problems in Figure 8.

assignments (and hence renders inadmissable the present solution) and for which the union of the subsets contains at least one* feasible quadratic assignment which is optimal for the set being subdivided.

Figure 12 shows all conflicts in the optimal assignment solution of Figure 10(a) arising between *pairs* of the assignments. Any of these conflicts could be used as the basis for subdividing. For example, Figure 13(a) illustrates subdivision of the basis of the conflicts between the assignments ($AC24$) and ($AD13$). Of course, in the optimal linear assignment at the resulting nodes there may persist conflicts which were present at the parent node: thus at node ($\overline{AD13}$) it might be necessary to resolve at a next level the conflict between ($AC24$) and ($BD23$) should it be present in the new solution at that node, as illustrated in Figure 13(b). On the other hand, the conflict may not appear in subsequent linear assignment solutions and hence not have to be considered explicitly.

More stringent subdivisions may result, perhaps, by using more than a single conflict. Letting $\overline{AC24}$ denote the event "not assignment $AC24$", $AC24$ the event "assignment $AC24$", "$\oplus$" logical "or" (disjunction) and "$\cdot$" logical "and" (conjunction), we can represent the resolution of the conflict between $AC24$ and $AD13$, for example, as simply:

$$\overline{AC24} \oplus \overline{AD13}, \tag{15}$$

meaning simply that a necessary condition for feasibility is the event "not assignment $AC24$" or "not assignment $AD13$," or both. Suppose we now consider the conflict between, say, $AC24$ and $BD23$. For resolution of this conflict we must have $\overline{AC24} \oplus \overline{BD23}$ so that in conjunction with (15) we must have for resolution of both:

$$(\overline{AC24} \oplus \overline{AD13}) \cdot (\overline{AC24} \oplus \overline{BD23}) = \overline{AC24} \oplus (\overline{AD13} \cdot \overline{BD23}) \tag{16}$$

*The subsets need not be mutually exclusive.

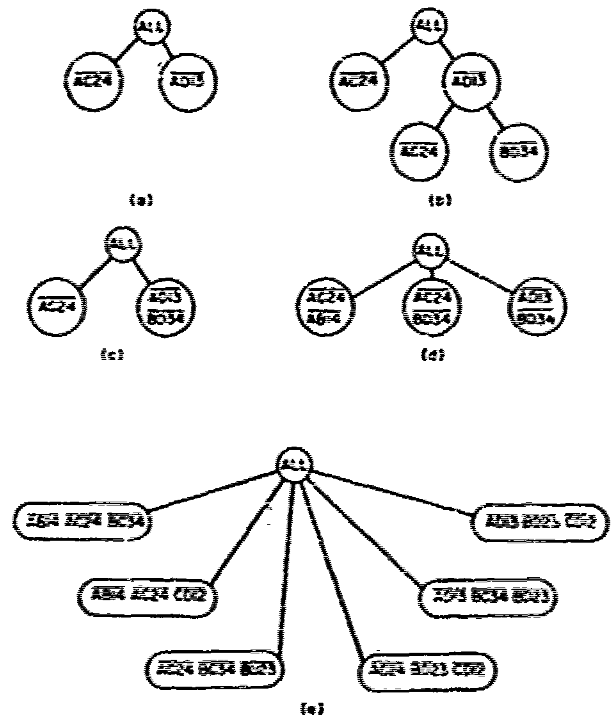


FIGURE 13. Illustrative, alternate ways to partition set of quadratic assignment solutions into subsets in pair-exclusion algorithms.

as represented in Figure 9(c). If desired we could consider in conjunction with these two, say, the pair of conflicts $AD13$ and $BC34$, with the result:

$$(17)\qquad (\overline{AC24}\oplus\overline{AD13})\cdot(\overline{AC24}\oplus\overline{BD23})\cdot(\overline{AD13}\oplus\overline{BC34})$$

$$=\overline{AC24}\cdot\overline{AD13}\oplus\overline{AC24}\cdot\overline{BC34}\oplus\overline{AD13}\cdot\overline{BD23}$$

as represented in Figure 13(d), or perhaps these in conjunction with the pair $BC34$ and $CD12$ as represented in Figure 13(e). Similarly, any combination of the constraints can be considered.†

For the resulting nodes we proceed just as before to determine an optimal linear assignment, where now every assignment appearing in the expression defining a node is made inadmissable. For a lower bound for a node probably the easiest is to simply use the largest alternate cost of the assignments to be excluded at the node. A more stringent bound of course would result by actually making the elements inadmissable and reducing the resulting assignment matrix or by obtaining an optimal assignment solution.

These illustrations serve to indicate the nature of the choices to be made at a node in the tree. Referring to Figure 13 it is clear that (c) is preferable to (b) since the resulting subdivision is obtained at a single level of the tree, and with no increase in the number of nodes at that level. Similarly it can

†Note that conflicts involving assignments not actually committed in the optimal linear assignment can be used in conjunction with these as well. Thus, for example, since the cost of assigning AB to 23 in Figure 6(a) is zero this could well appear as an assignment in the solution at a subsequent level. Since this assignment, however, would conflict with, say, assignment $CD12$, one could if desired explicitly consider that conflict at the present node in forming the subdivisions.

be argued that the subdivision in Figure 13(e) is preferable to that on the first level in Figure 11 since the total number of subdivisions or nodes to be considered is the same while the size of each of the subsets in the former is smaller. Unclear, however, is the choice among, say, those in Figures 13(c), (d), and (e), a choice involving a larger number of subsets, but each of smaller size. On the one hand it is necessary to determine a bound and/or optimal assignment for each node, but on the other, the subset being smaller the greater is the possibility the node will be bounded by an existing feasible solution and hence not require any further consideration. Similarly with regard to the evaluation of lower bounds at a node: reducing the assignment matrix and/or determining an optimal assignment yields a more stringent bound and enhances the likelihood that the node will be bounded, but to do either requires establishing and manipulating the appropriate assignment matrix for that node in contradistinction to the use of the alternate cost information which requires no explicit consideration of that node's matrix. These choices remain subjects for empirical study.

In concluding discussion it is noted that in practice it may be efficient within this exclusion type of algorithm to be looking at each node for mandatory assignments as well as for assignments to be excluded. As before, this could be done simply by checking the alternate costs of the assignments which occur in the optimal linear assignment. For each mandatory assignment discovered the appropriate related assignments would at that time be made inadmissable, thereby making subsequent bounding and search potentially more effective.

As an illustration, we again consider the problem in Figure 8, and solve it with the following algorithm. For a search strategy we use the same level-by-level strategy used for illustration throughout, choosing at each subdivision an unexplored node having the smallest lower bound. In each optimal linear assignment the alternate costs of the assignments are all checked to see if the assignment can be shown to be mandatory. Whenever a node is encountered for which the optimal linear assignment results in a nonfeasible quadratic assignment, a subdivision is formed in the following way. All pairs of assignments in the optimal solutions are investigated for conflict and those so found are noted together with their alternate costs (as was done in Figure 12). From this list is selected the pair for which the smaller of the two alternate costs is largest among the minimum of all pairs: ultimately every feasible quadratic assignment which resolves all of these conflicts must have a cost at least as large as this value. Should there be more than one pair with this same minimum, a pair is selected for which the other alternate cost is maximum. (In Figure 12 we thus select either the pair *AC*24–*AD*13 or *AC*24–*BD*23.) For the selected pair, we then search for other pairs which have an assignment identical to the assignment in the selected pair having the larger alternate cost, and use these pairs in conjunction with the selected pair. (In Figure 12 we would thus form from the pairs *AC*24–*AD*13 and *AC*24–*BD*23 the subdivision: $(\overline{AC24} \oplus \overline{AD13} \cdot \overline{BD23})$.) The resulting subdivision will thus insure that at least the minimal increase in cost that eventually must be incurred will in fact be incurred now, and possibly a greater increase—but without proliferating the number of individual subsets to be considered at this level in the tree. Finally, other assignments are sought having exactly the same conflicting assignments as this assignment in the original selection pair with the higher alternate cost (in the example, all assignments having the conflicts with the same assignments *AD*13 and *BD*23 as does *AC*24) and these conjugated with the present set of conflicts (there are no such assignments in the example). The result is a subdivision consisting of two subsets.

In the event there is conflict in an optimal linear assignment solution, but no conflict among simple *pairs* of assignments we simply choose the first subset of the assignments discovered to be in conflict, remove the assignments in the subset not contributing to the conflict, and subdivide on the basis of

e remaining assignments—each assignment defining one subdivision.

Occasionally in the development of the enumeration tree it can be shown at a node that a particular pair of assignments is mandatory in the same sense as in Section III. For example, in Figure 14(a) we have a solution 445 *CL* labelled with a^* which was obtained as follows: The solution to the linear assignment problem obtained after adding $\overline{AD32}$ had a reduction of 445 (equal to the current best feasible solution). The alternate cost of assignment *CD*12 was 19, and therefore *CD*12 would have to be in any optimal solution to the problem. Similarly the updated alternate costs of *AB*34 and *BD*13 force them into a solution. These assignments taken jointly give $A=4$, $B=3$, $C=2$, and $D=1$ for an actual cost of 445. It is interesting to note that the assignments which are obtained in this manner may result in a nonfeasible solution to the linear assignment problem at that point in the enumeration tree. For example, we might have arrived at the above solution even if *BC*23 were specifically excluded.

For the Gavett and Plyter problem in Figures 3 and 8, the tree which is elaborated is shown in Figure 14. At least for this problem the optimal linear assignment solution is not "close" to the optimal quadratic assignment solution.

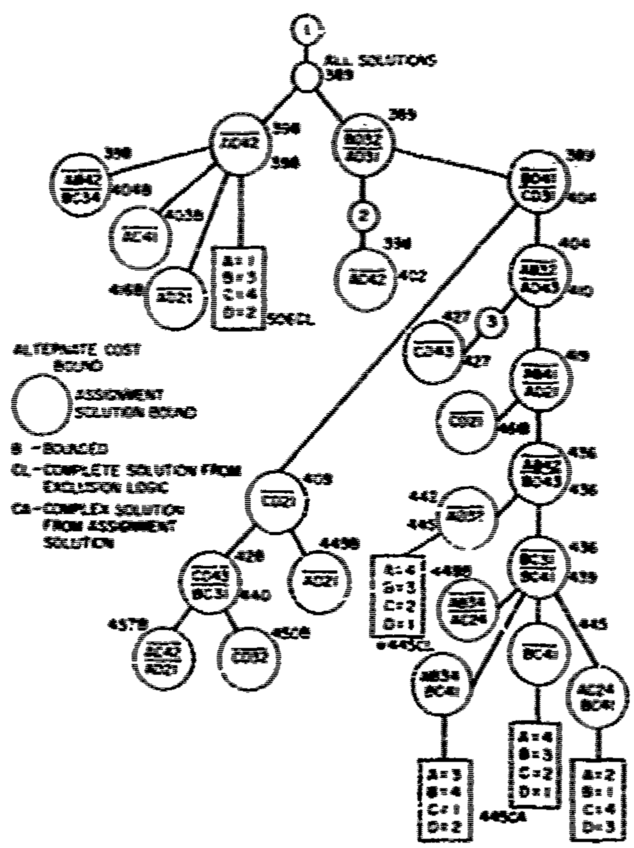
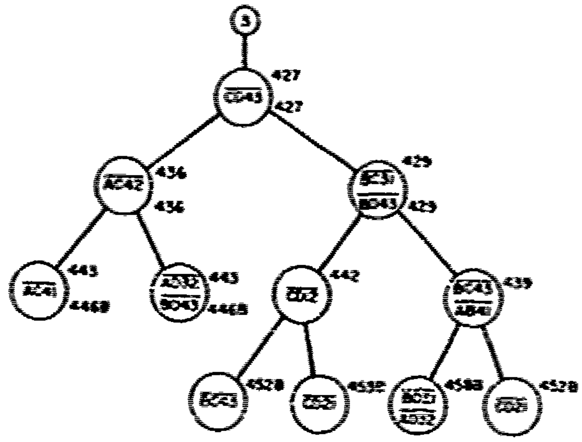
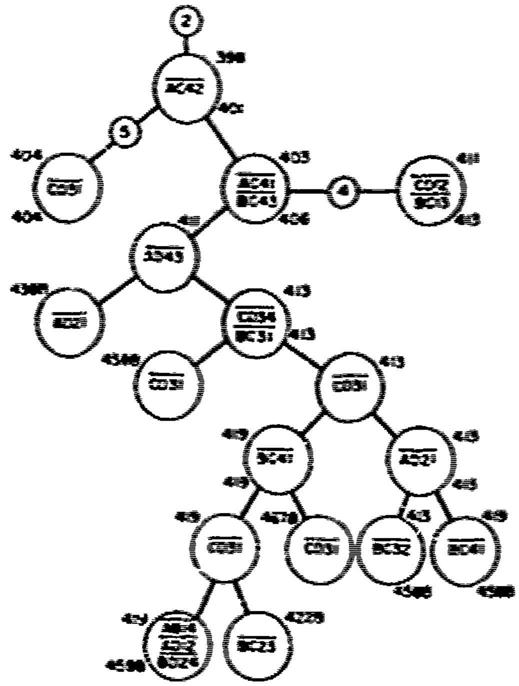


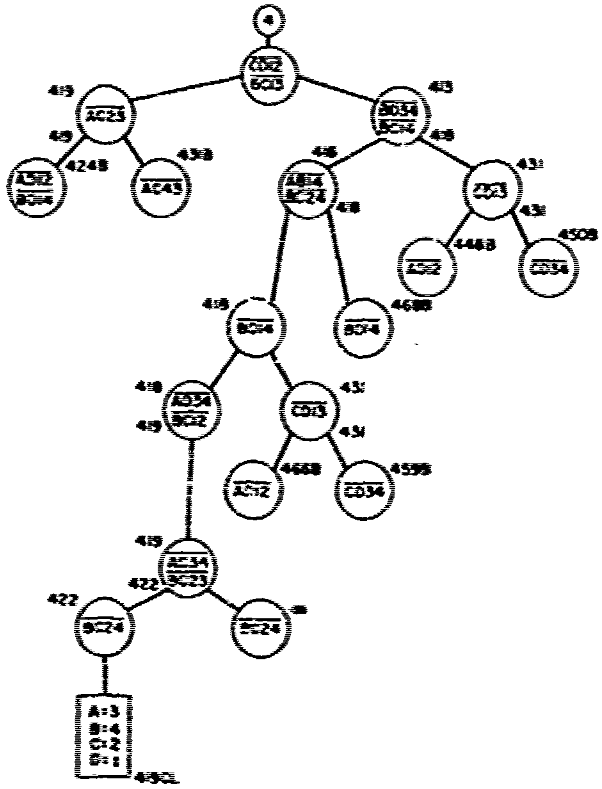
FIGURE 14 Tree elaborated for problem of Figure 3 with illustrative pair-exclusive algorithm.

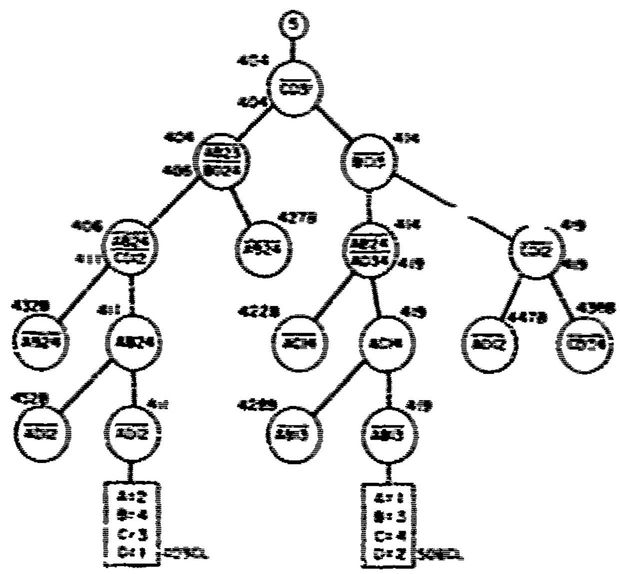
---

*Were he to present constraints involving, say, triplets of assignments, one might then wish to formulate the problem in terms of an $(n(n-1)(n-2)/6)\times(n(n-1)(n-2)/6)$ linear assignment problem, and proceed with a triplet-assignment or triplet-exclusion algorithm akin to algorithms discussed in Sections IV and V.









VI. CONCLUDING REMARKS

In this paper three classes of algorithms have been discussed for solving quadratic assignment problems. Regardless of its class each algorithm which has been considered is reliable in the respect that if carried to completion it guarantees the discovery of an optimal solution. Furthermore in finding an optimal solution it proceeds first to a feasible solution and then to better and better feasible solutions so that, if desired, problem-solving can be terminated prematurely with a usable, if not optimal, solution. In addition, these procedures can all efficiently exploit information available beforehand regarding the value of a known feasible solution and hence, for example, can be readily used in conjunction with heuristic procedure which gives good sub-optimal solutions. Moreover, if desired, all of the algorithms discussed can be used with slight modification to determine *all* optimal solutions, or a specified number of the most preferred solutions, and so on.

Common to all three classes of algorithms is the structuring of the quadratic assignment problem in terms of a related linear assignment problem. In each this latter problem is then used in directing the tree-search process and in bounding and dominance considerations designed to reduce search. However, between the linear assignment problems used for the single-assignment algorithms and the pair-assignment algorithm there are major differences.

In the single-assignment case the linear assignment problem is only of dimensions $n \times n$ and has the property that a feasible solution always represents a feasible solution to the quadratic assignment problem. Its shortcoming, however, lies in the fact that the cost structure embodied in the linear assignment problem is not, in general, an exact representation of the true cost structure of the quadratic problem, but only an approximation. The effect is to diminish the stringency of the bounds and dominance tests, and to necessitate the periodic expenditure of problem-solving time in updating the representation of the cost structure.

In the pair-assignment or pair-exclusion cases, on the other hand, the associated linear assignment problem is significantly larger, being of dimensions $n(n-1) \times n(n-1)/2$, but in this larger problem it is possible to represent exactly the cost structure of the quadratic problem. However, the shortcoming of this representation lies in the fact that a feasible solution to the linear problem need not constitute a feasible solution to the quadratic problem.

Even from our experience with the one sample problem in this paper it is clear that the different algorithms can give rise to the elaboration of quite different partial trees of solutions with quite differing numbers of nodes (see Figures 4–7, 9, and 14). However, in light of the fact that the time required to elaborate and evaluate a single node in a tree can differ markedly among the algorithms, it is difficult to assess the relative efficiency of the different algorithms even for this one, single problem. In practice, moreover, the relative efficiency may well turn out to be highly dependent on the particular form of the quadratic assignment being solved. For example, were the coefficients c_{ij} in a problem with objective function (5) to predominate, the approximate cost structure in the single-assignment methods might in fact be quite "close" to the true cost structure and hence that class of algorithms be quite efficient. On the other hand, were an architect to pose a problem with a large number of pairwise constraints on permissible assignments, the ability to reflect directly their implications in the cost representation of the pair-assignment or pair-exclusion classes might render these approaches more efficient. Hopefully, it will be possible to glean information pertaining to these questions from the computational results to be reported in the subsequent paper.

REFERENCES

[1] Armour, G. C. and E. S. Buffa, "A Heuristic Algorithm and Simulative Approach to Relative Location of Facilities," Management Science *9*, 294–309 (Jan. 1963).

[2] Breuer, M. A., "The Formulation of Some Allocation and Connection Problems as Integer Programs," Nav. Res. Log. Quart. *13*, 83–95 (Mar. 1966).

[3] Conway, R. W. and W. L. Maxwell, "A Note on the Assignment of Facility Location," J. Ind. Eng. *12*, 34–36 (Jan.–Feb. 1961).

[4] Eastman, W. L., "Linear Programming with Pattern Constraints," Ph. D. Thesis, Computation Laboratory Report No. BI. *20*, Harvard University (1958).

[5] Flood, M. M., "The Traveling-Salesman Problem," Operations Research, *4*, 61–75 (Feb. 1956).

[6] Ford, L. R., Jr. and D. R. Fulkerson, Flows in Networks (Princeton University Press, New Jersey, 1962).

[7] Gaschutz, G. K. and J. H. Ahrens, "Suboptimal Algorithms for the Quadratic Assignment Problem," Nav. Res. Log. Quart. *15*, 49–62 (Mar. 1968).

[8] Gavett, J. W., and N. V. Plyter, "The Optimal Assignment of Facilities to Locations by Branch and Bound," Operations Research *14*, 210–232 (Mar.–Apr. 1966).

[9] Geoffrion, A., "Integer Programming by Implicit Enumeration and Balas' Method," SIAM Review *9*, 178–190 (1967).

[10] Gilmore, P. C., "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem," J. Soc. Ind. and Appl. Math. *10*, 305–313 (June 1962).

[11] Golomb, S. W. and L. D. Baumert, "Backtrack Programming," J. Assoc. for Computing Machinery *12* 516–524 (Oct. 1965).

[12] Graves, G. W. and A. B. Whinston, "An Algorithm for the Quadratic Assignment Problem," Working Paper No. 110, Western Management Service Institute, UCLA, Nov. 1966.

[13] Hillier, F. S., "Quantitative Tools for Plant Layout Analysis," J. Ind. Eng. *14*, 33–40 (Jan.–Feb. 1963).

[14] Hillier, F. S. and M. M. Connors, "Quadratic Assignment Problem Algorithms and the Location of Indivisible Facilities," Management Science *13*, 42–57 (Sept. 1966).

[15] Koopmans, T. C. and M. Beckmann, "Assignment Problems and the Location of Economic Activities," Econometrica *25*, 53–76 (Jan. 1957).

[16] Land, A. H., "A Problem of Assignment with Interrelated Costs," Operational Research Quarterly *14*, 185–198 (June 1963).

[17] Land, A. H. and A. Doig, "An Automatic Method of Solving Discrete Programming Problems," Econometrica *28*, 497–520 (July 1960).

[18] Lawler, E. L., "The Quadratic Assignment Problem," Management Science 9, 586–599 (July 1963).

[19] Lawler, E. L., "Notes on the Quadratic Assignment Problem," Harvard Computation Laboratory, unpublished paper (Apr. 1960).

[20] Little, J. D. C., K. G. Murty, D. W. Sweeney and K. Caroline, "An Algorithm for the Traveling Salesman Problem," Operations Research *11*, 972–989 (Nov.–Dec. 1963).

[21] Markowitz, H. M. and A. S. Manne, "On the Solution of Discrete Programming Problems," Econometrica *25*, 84–110 (Jan. 1957).

[22] Maxwell, W. L., "The Scheduling of Single Machine Systems: A Review," The International Journal of Production Research *3*, 177–199 (1964).

[23] Nugent, C. E., T. E. Vollman, and J. Ruml. "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations." Operations Research *16*, 150–173 (Jan.–Feb. 1968).

[24] Pegels, C. C., "Plant Layout and Discrete Optimizing." International Journal of Production Research *5*, 81–92 (1966).

[25] Pierce, J. F. and D. J. Hatfield, "Production Sequencing by Combinatorial Programming," Operations Research and the Design of Management Information Systems (J. F. Pierce, Ed.) (Tech. Association of the Pulp and Paper Industry, New York, 1967).

[26] Rossman, M. J. and R. J. Twery, "Combinatorial Programming." Unpublished paper presented at 6th Annual meeting of Operations Research Society of America, Boston, 1958.

[27] Rutman, R. A., "An Algorithm for Placement of Interconnected Elements Based on Minimum Wire Length." Proc. AFIPS Spring Joint Computer Conference (1964), p. 477–491.

[28] Shapiro, D., "Algorithms for the Solution of the Optimal Cost Traveling Salesman Problem." Sc. D. Thesis, Washington University, St. Louis, 1966.

[29] Sprague, C. R., "Solution of Three Related Combinatorial Problems by Network-Flow and Branch and Bound Methods." Sloan School Working Paper No. 451–70, M.I.T. (Mar. 1970).

[30] Steinberg, L., "The Backboard Wiring Problem: A Placement Algorithm," Soc. for Ind. and Appl. Math. *3*, 37–50 (Jan. 1961).

[31] Vollman, T. E. and E. S. Buffa, "The Facilities Layout Problem in Perspective," Management Science *12*, 450–468 (June 1966).

[32] Whitehead, B. and M. Z. Elders, "An Approach to the Optimum Layout of Single-Story Buildings," Architect's Journal *139*, 1373–1380 (June 1964).

[33] Wimmert, R. J., "A Mathematical Model of Equipment Location," Journal of Industrial Engineering 9, 498–505 (Nov.–Dec. 1958).

OPTIMAL INTERDICTION POLICY FOR A FLOW NETWORK

P. M. Ghare, D. C. Montgomery, and W. C. Turner

Department of Industrial Engineering
Virginia Polytechnic Institute

ABSTRACT

This paper analyzes the problem faced by a field commander who, confronted by an enemy on N battlefields, must determine an interdiction policy for the enemy's logistics system which minimizes the amount of war material flowing through this system per unit time. The resource utilized to achieve this interdiction is subject to constraint. It can be shown that this problem is equivalent to determining the set of arcs Z^* to remove subject to constraint from a directed graph G such that the resulting maximal flow is minimized. A branch and bound algorithm for the solution to this problem is described, and a numerical example is provided.

I. INTRODUCTION

Consider a field commander who is simultaneously confronted by an enemy on N battlefields. This enemy has the capability to supply his forces through a logistics support system emanating from M depots and ultimately connecting with these battlefields. The field commander has available a limited number of attack resources (e.g., aircraft) with which he may assault the enemy logistics system. We now assume that this logistics system is a network of routes from the M depots, passing through L intermediate junctions, and terminating at the N battlefields. Our field commander wishes to allocate his attack resources optimally, that is, so that after assaulting the logistics system the enemy's maximal flow of war material per unit time is minimized. Certain assumptions regarding this tactical situation may be stated formally:

1. There is a resource cost of assaulting a supply route, and this cost is known with certainty.

2. Every supply route has a maximal flow per unit time that it can sustain; it is called the capacity of the route.

3. All attacks are successful with probability one, and partial destruction of a route is impossible.

4. The destruction of depots, intermediate junctions, or battlefields is prohibitively costly and cannot be considered.

5. The enemy has available a technique for determining a policy which yields the maximal amount of war material flowing through the network per unit time.

The logistics system under study may be thought of as a graph, G, if we let the supply routes be arcs and the depots, intermediate junctions, and battlefields be vertices. As every arc would be incident with two vertices and every vertex would have at least one arc incident with it, this would also be a connected graph. Finally, we define a flow network to be a connected graph in which every arc has an orientation and a capacity. This orientation categorizes the depots as "sources," as all arcs are incident from them, and the battlefields as "sinks," as all arcs are incident to them.

II. PROBLEM STATEMENT

In accordance with the above discussion and assumptions, the original problem can now be restated as follows: "Given a graph, G, which defines a flow network and limited resources, determine the set of arcs, Z^*, to remove from G such that the resulting maximal flow is minimized." An obvious constraint is that the total resource expenditure required to obtain Z^* must be less than or equal to the total resource availability. Employing a symbolism similar to that of Berge [2], this problem may be stated mathematically as

$$\text{maximize } \pi = \phi(G) - \phi(G - Z^*),$$

subject to the constraint

$$\sum_{i \in Z^*} R_i \leq R_{\max},$$

where R_i = the resource expenditure necessary to destroy arc i,
$R_{\max}$ = the maximum resource availability, and
$\phi(X)$ = the maximal flow through network X.

Other authors have considered similar problems. Bellmore et al. [1] utilize a dynamic programming technique to determine which links to remove from a communications network. The flows in all arcs of this network are constrained to be either zero or one. Wollmer [6] proposes two techniques for determining the most vital link in a railway network. The first technique considers the case of an unlimited number of trains, while the second assumes that the railway has a fixed number of trains. Both procedures allow removal of only a single arc. In another work Wollmer [7] considers the removal of n arcs, but no resource expenditure is necessary for removal. If we set $R_i = 1$ for all arcs in G, and $R_{\max} = n$ then the problem analyzed herein is identical to that of Wollmer.

III. THEORETICAL CONSIDERATIONS

The development of this section is extracted from our previous paper [4]. Network flow has been examined extensively by Ford and Fulkerson [3], and they have provided the following fundamental theorem concerning maximal flow:

THEOREM 1: The maximal flow from a set of sources $\bar{S}$ to a set of sinks $\underline{S}$ in a flow network, G, is equal to the minimum of the capacities of the cutsets of arcs separating $\bar{S}$ and $\underline{S}$.

This theorem forms the basis of a well-known algorithm by Ford and Fulkerson for determining the maximal flow. It is assumed that the enemy will use this procedure to determine his maximal flow policy. Theorem 1 leads to Corollary 1.

COROLLARY 1: The maximal flow through G is an univalued function of the arc capacities and the network configuration.

COROLLARY 2: It is possible to control (or alter) this maximal flow by manipulating either the arc capacities and/or the network configuration. We shall be concerned only with the manipulation of arc capacities.

Consider now an arc i included in the graph G. If the capacities of all other arcs remain unchanged then $\phi(G)$ is, by Corollary 1, a univalued function of the capacity of arc i. Let C_i represent the capacity of arc i. If $\phi(0)$ and $\phi(\infty)$ represent the maximal flow when $C_i = 0$ and $C_i = \infty$, respectively, then the quantity $C_i^* = \phi(\infty) - \phi(0)$ could be called the critical capacity of arc i. It can be argued that so long as

$C_i \leq C_i^*$, arc i is included in at least one minimal cutset. If $C_i > C_i^*$, then arc i would not be included in any minimal cutset and would contain a "slack capacity." Let $I(h_i)$ be the "interdiction" caused by the perturbation h_i in the capacity of arc i. Thus

$$\phi(C_i + h_i) = \phi(C\cdot) + I(h_i),$$

and

$$(1) \qquad I(h_i) = \begin{cases} h_i & \text{if } h_i < o \text{ and } |h_i| < C_i \\ -c_i & \text{if } h_i < o \text{ and } |h_i| > C_i \\ h_i & \text{if } h_i > o \text{ and } C_i + h_i < C_i^* \\ C_i^* - C_i & \text{if } C_i + h_i \geq C_i^* \text{ and } h_i > 0. \end{cases}$$

If the perturbation h_i required R_i units of resource then R_i would be considered as the cost of the interdiction $I(h_i)$.

This definition holds only for arcs included in a minimal cutset. An arc which is not included in such a cutset will, by the above reasoning, have a slack capacity. If $\omega^+(V)$ and $\omega^-(V)$ represent the sets of arcs incident from and to vertex V, respectively, then slack exists in arc

$$i \epsilon \omega^+(V) \text{ if } \sum_{k\epsilon\omega^-(V)} C_k' < \sum_{j\epsilon\omega^+(V)} C_j.$$

By similar reasoning, slack exists in arc $i\epsilon\omega^-(V)$ if $\sum_{j\epsilon\omega^+(V)} C_j' < \sum_{k\epsilon\omega^-(V)} C_k$. The quantity C_i' is the effective capacity of arc i. We may now define the slack associated with arc i, σ_i, as

$$(2) \qquad \sigma_i = \begin{cases} \sum_{j\epsilon\omega^+(V)} C_j - \sum_{k\epsilon\omega^-(V)} C_k' | i\epsilon\omega^+(V), \ \sigma_i \geq o \\ \sum_{k\epsilon\omega^-(V)} C_k - \sum_{j\epsilon\omega^+(V)} C_j' | i\epsilon\omega^-(V), \ \sigma_i \geq o \end{cases}$$

and $c_i' = C_i - \sigma_i$.

Shapley [5] has investigated the interactive effect of capacity variation for two arcs in a network. This interaction may be either assistance or hindrance, and is defined by the difference quotient

$$(3) \qquad q_{ij} = \{\phi(C_i + h_i; C_j + h_j) - \phi(C_i + h_i; C_j) - \phi(C_i; C_j + h_i) + \phi(C_i; C_j)\}/\{h_i \cdot h_j\},$$

where $\phi(\cdot\ ;\ \cdot)$ is the maximal flow through the network, $C_i + h_i$ and $C_j + h_j$ are nonnegative, and $h_i \cdot h_j$ is nonzero. If the difference quotient is positive, then arcs i and j assist each other, and if the difference quotient is negative then arcs i and j hinder each other. If arcs i and j have $q_{ij} > o$ and $\sigma_i > o$, the slack could be reduced by increasing the capacity of arc j. Further, it would be possible to reduce the slack in arc i to zero by reducing the capacities of arcs having a negative difference quotient with i. To identify arcs with positive and negative difference quotients we quote without proof the following theorem due to Shapley:

THEOREM 2: (a) If the terminal node of i is the initial node of j, then $q_{ij} \geq o$; (b) If i and j have the same initial (terminal) node $q_{ij} \leq o$; (c) If the initial node of i is a source and the terminal node of j a sink $q_{ij} \geq o$.

When the slack of any arc is reduced to zero that arc would now be included in a minimal cutset. Therefore, it is now possible to define the interdiction function for all arcs in the following manner.

(1) If arc i is included in a minimal cutset Equation (1) gives the interdiction function $I(h_i)$.

(2) If arc i is not included in a minimal cutset, but can have its slack σ_i reduced to zero by altering the capacities of other arcs, then the interdiction function $I(h_i)$ is the variation in the maximal flow caused by a variation of h_i in arc i after the alternations for the removal of the slack.

(3) If arc i is not included in a minimal cutset and the slack σ_i cannot be reduced to zero, the $I(h_i)=0$.

The resource cost of interdiction in cases (1) and (3) would be the cost of the capacity variation h_i only, but for case (2) the cost of capacity variation necessary to reduce the slack to zero would also have to be included.

From the above definition, we see that the interdiction function possesses the following properties:

$$|I(h_i)| \leq |h_i|. \tag{4}$$

$$I(h_i+h_j)=I(h_j+h_i). \tag{5}$$

$$|I(h_i+h_j)| \leq |I(h_i)|+|I(h_j)|. \tag{6}$$

IV. STATEMENT OF THE ALGORITHM

For the problem under consideration the capacity variation h_i would be exactly equal to $-C_i$ as arcs cannot be partially destroyed. Hence the problem may now be restated as find Z^* to minimize

$$\pi=\phi(G)+I\left(\sum_{i\in Z^*}\{-C_i\}\right)$$

subject to the constraint

$$\sum_{i\in Z^*} R_i \leq R_{\max}.$$

The solution can be obtained through implicit enumeration using a branch and bound procedure. The decisions either to include or exclude a particular arc form the branching procedure. The bounds on the interdiction can be obtained by considering the following:

$$\pi=\phi(G)+I\left(\sum_{i\in Z}\{-C_i\}\right).$$

Therefore, by Equation 6

$$\pi \geq \phi(G)+\sum_{i\in Z} I(-C_i).$$

Continuing,

$$\pi \geq \phi(G)+\sum_{i\in Z}\left\{\frac{I(-C_i)}{R_i}\cdot R_i\right\}$$

$$\pi \geq \phi(G)+\min_{i\in Z}\left\{\frac{I(-C_i)}{R_i}\right\}\sum_{i\in Z} R_i$$

$$\pi \geq \phi(G)+\min_{i\in Z}\left\{\frac{I(-C_i)}{R_i}\right\} R_{\max}. \tag{7}$$

Hence Equation 7 is a lower bound on the maximal flow through G subject to resource availability R_{max}. A computational algorithm for solving the problem may now be described.

In addition to the notation used previously let Z_k be the set of arcs destroyed at iteration k and T_k be the interdiction achieved after k iterations. Now the algorithm may be stated as follows: Begin with $k=1$, $T_1=\phi$, $Z_1=\phi$, $D_1=R_{max}$. For the node "All Arcs" set $UB_0(G)=0$. Create the dummy source $\tilde{S}$ and the dummy sink $\underset{\sim}{S}$ (if necessary).

STEP 1: Lable those vertices nearest $\tilde{S}$ according to

$$I_V=+\sum_{i\epsilon\omega^+(V)} C_i$$

Set $C_i'=C_i$ for all $i\epsilon\omega^+(V)$.

STEP 2 (A): Proceed to the next set of vertices and label them according to the following rules:

$$I_V=\begin{cases} +\sum\limits_{i\epsilon\omega^+(V)} C_i \;\Big|\; \sum\limits_{i\epsilon\omega^+(V)} C_i < \sum\limits_{j\epsilon\omega^-(V)} C_j' \\ -\sum\limits_{i\epsilon\omega^+(V)} C_j' \;\Big|\; \sum\limits_{i\epsilon\omega^-(V)} C_i' < \sum\limits_{j\epsilon\omega^-(V)} C_j \\ \pm\sum\limits_{i\epsilon\omega^-(V)} C_i', \text{ otherwise.} \end{cases}$$

(B) If $I_V<0$ compute $\theta=\sum\limits_{j\epsilon\omega^-(V)} C_j-|I_V|$.

set $C_i'=C_i-\theta \mid i\epsilon\omega^+(V)$.

(C) If $I_V>0$ set $C_i'=C_i \mid i\epsilon\omega^+(V)$.

STEP 3: Repeat Step 2 until the vertices nearest $\underset{\sim}{S}$ have been labeled, then proceed to Step 4.

STEP 4 (A) Starting with the arcs nearest $\underset{\sim}{S}$ and proceeding backwards consider those with $I_V=+$. For these arcs compute

$$C_i'=C_i-\theta.$$

where

$$\theta=\sum_{j\epsilon\omega^-(V)} C_j'-|I_V|.$$

(B) For all arcs having slack $\sigma_i>0$ (σ_i as defined by Equation (2)) identify those having a negative difference quotient. Among these determine the minimal cost of removing the slack. Then set C_i' = the interdiction obtained and R_i = the least cost set of variations to remove the slack σ_i.

STEP 5: Repeat Step 4 until no further capacities can be adjusted. Go to Step 6.

STEP 6: Compute the quantities

$$\delta_i = \frac{C_i'}{R_i} \mid i \in G.$$

Select

$$\delta_r^* = \max_{i \in G} \{\delta_i \mid R_i \leq D_k\}$$

and

$$\delta_n' = \max_{n \in G} \{\delta_i \mid R_i \leq D_k, \quad n \neq r\}.$$

STEP 7: Create the nodes on the tree (r) and $(\bar{r})$. For node $(\bar{r})$ compute the upper bound:

$$UB_k(\bar{r}) = \delta_n' D_k + T_{k-1}.$$

STEP 8: To Branch from node (r), we set $k = k+1$

$$D_k = D_{k-1} - R_r,$$

$$T_k = T_{k-1} + C_r',$$

$$Z_k = Z_{k-1} \cup r, \text{ and}$$

$$G = G \cap \bar{r}.$$

Return to Step 1 and continue, initially setting all C_i to their original values, until $\min_{i \in G} \{R_i\} > D_k$. Denote T_k as T^*. Go to Step 9.

STEP 9: Search back up the tree until a node $(\bar{j})$ is encountered such that $UB_m(\bar{j}) > T^*$. If more than one $UB_m(\bar{j}) > T^*$ branch from the maximum. Begin branching from that node by returning to the Step 6 calculation at which arc j was selected for interdiction.

STEP 10: Select for interdiction the arc designated by δ_n' and repeat Steps 6–8 until one of two possible events occur:

(a) $\min_{i \in G} \{R_i\} > D_k$. Go to Step 11.

(b) The arc i is selected for interdiction, and a decision $(\bar{i})$ is incorporated in this branch previously. Go to Step 12.

STEP 11: If $T_k > T^*$ then set $T^* = T_k$ and go to Step 12.

STEP 12: Continue to search back up the tree for a node with $UB_m(\bar{j}) > T^*$, repeating Steps 9–12 until no further branches need be searched. Then T^* is the optimal interdiction and the set of arcs to remove is given by $Z_k = Z^*$.

Steps 1 through 5 of this algorithm compute the effective capacities for each arc in the network. The ratios formed in Step 6 provide a measure of the interdiction obtained per unit of resource expended. Naturally, as one wishes to maximize this interdiction, the largest ratio is chosen. The upper bound for not selecting this arc would be the product of the next highest ratio and the available resource, plus the interdiction obtained to that stage. Steps 10–12 are concerned with searching back up the decision tree, and branching from all nodes whose upper bound is greater than the current maximal interdiction. Dynamic programming could also have been utilized to solve this problem.

V. NUMERICAL EXAMPLE

To illustrate the computations consider the following numerical example. Suppose a field commander is engaged on three battlefields by an enemy who supplies his forces from four depots. The enemy transportation is shown in Figure 1. Initially 720 truckloads per day of supplies can be delivered

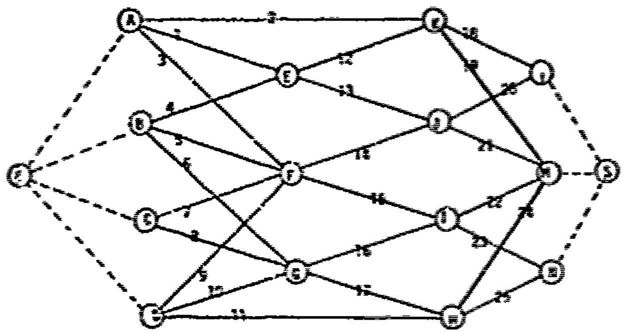


FIGURE 1. The initial network

through this network. The capacities of the routes in truckloads of material per day and the expected aircraft losses are given in Table 1. Fifteen aircraft are available to attack the supply system.

TABLE 1. *Network Configuration Data*

Arc	Capacity (truckloads per day)	Cost to Destroy (aircraft)	Arc	Capacity (truckloads per day)	Cost to Destroy (aircraft)
1	60	5	14	120	4
2	70	4	15	150	6
3	60	5	16	120	6
4	50	3	17	80	4
5	50	3	18	80	4
6	60	5	19	50	5
7	100	3	20	100	5
8	80	5	21	80	4
9	50	5	22	180	6
10	100	5	23	100	4
11	80	4	24	80	5
12	60	4	25	100	6
13	60	7			

The decision tree which illustrates the steps needed to solve this problem is shown in Figure 2.

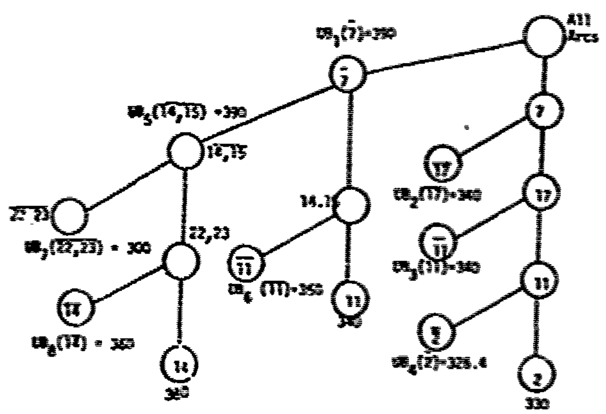


FIGURE 2. Decision tree depicting problem solution

TABLE 2. *Step Six Ratios For Each Iteration*

arc	Iteration 1	2	3	4	5	6	7	8
1	12.0	12.0	12.0	0	12.0	12.0	12.0	0
2	17.5	17.5	17.5	17.5*	17.5	17.5	17.5	17.5
3	12.0	12.0	12.0	12.0	12.0	0	12.0	0
4	16.6	16.6	16.6	16.6	16.6	16.6	16.6	16.6
5	16.6	16.6	16.6	16.6	16.6	0	16.6	0
6								
7	33.3				33.3	0	33.3	0
8								
9	10.0	10.0	10.0	0	10.0	0	0	0
10								
11	20.0	20.0	20.0*		20.0	20.0*	20.0	20.0
12								
13								
14						0		30.0*
15						0		0
16	20.0	20.0	20.0	0	20.0	20.0	20.0	0
17	20.0	20.0*			20.0	20.0	20.0	20.0
18								
19								
20								
21								
22								
23								
24								
25								
6, 10	12.0	12.0	1.0	0	12.0	12.0	12.0	0
8, 10	14.0	14.0	6.0	0	14.0	14.0	14.0	0
12, 13	10.0	10.0	10.0	0	10.0	10.0	10.0	0
14, 15	26.0	16.0	16.0	0	26.0*	0	26.0	0
18, 19	13.3	13.3	13.3	0	13.3	13.3	13.3	0
20, 21	17.6	6.7	6.7	0	17.6	5.6	17.6	0
22, 23	26.0	16.0	16.0	0	26.0	12.0	26.0*	0
24, 25	14.5	14.5	7.3	0	14.5	16.0	14.5	0

and the ratios calculated at Step 6 are displayed as Table 2. The asterisks in Table 2 denote the ratios chosen at each iteration. The optimal solution yields a final maximal flow of 340 truckloads per day, and is shown in Figure 3.

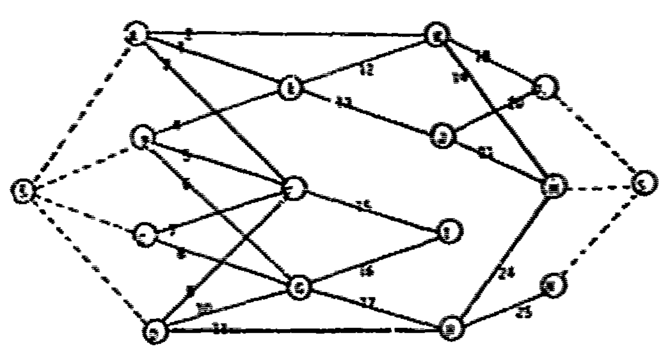


FIGURE 3. The optimal solution

REFERENCES

[1] Bellmore, M. H., Greenberg, and J. Jarvis, "Optimal Attack of a Communications Network," Thirty-Second National Meeting of the Operations Research Society of America, Chicago, Illinois, November 1967.

[2] Berge, C., *The Theory of Graphs and Its Applications* (John Wiley and Sons, New York, 1962).

[3] Ford, L. R. and D. R. Fulkerson, *Flows in Networks* (Princeton University Press, Princeton, New Jersey, 1962).

[4] Ghare, P. M. and D. C. Montgomery, "Flow Management in Transportation Networks," in OR69, Proceedings of the Fifth International Conference on Operations Research. (Tavistock Publications, 1970).

[5] Shapley, L. S., *On Network Flow Functions*, Memorandum RM-2338, The Rand Corporation, Santa Monica, California, 1959.

[6] Wollmer, R. D., *Some Methods for Determining the Most Vital Link in a Railway Network*, Memorandum RM-3321-ISA, The Rand Corporation, Santa Monica, California, 1963.

[7] Wollmer, R. D., "Removing Arcs From a Network," Operations Research, *12*, 934–940 (1964).

(0, 1) HYPERBOLIC PROGRAMMING PROBLEMS*

Pierre Robillard

Département d'informatique
Université de Montréal

ABSTRACT

In the first part of this paper we study the unconstrained (0, 1) hyperbolic programming problem treated in [1]. We describe a new algorithm for this problem which produces an optimal solution by scanning just once the set of fractions to be analysed. This algorithm shows better computing performance than the one described in [1].

In the second part we study the (0, 1) hyperbolic programming problem with constraints given by inequalities on nondecreasing pseudo-boolean functions. We describe a "branch and bound" type algorithm for this problem.

INTRODUCTION

Problems in (0, 1) hyperbolic programming arises from practical situations where, for example, one has to minimize the quotient of a linear function representing the cost of certain items by a linear function representing the produced amount of these items.

The problem in its fully generality consists of minimizing a function

$$\frac{a_0 + \sum_{i=1}^{n} a_i x_i}{b_0 + \sum_{i=1}^{n} b_i x_i}$$

a) $a_i \geq 0 \quad b_i \geq 0 \qquad i = 0, 1, 2, \ldots, n$
b) $x_i = \{0, 1\} \qquad i = 0, 1, 2, \ldots, n$
c) $h_i(x_1, x_2, \ldots, x_n) < d_i \qquad i = 1, 2, \ldots, k$

and $h_i(x_1, \ldots, x_n)$ are pseudoboolean functions.

In the first part of this paper we consider the unconstrained problem which was discussed in [1]. We describe a new algorithm which improves the one given in [1]. In part two, we consider the (0, 1) hyperbolic programming with the constraints given by inequalities on nondecreasing pseudo-boolean functions, and we describe a branch and bound algorithm for it.

*This research was supported by a grant of the National Research Council of Canada.

PART 1 – THE UNCONSTRAINED PROBLEM

1.1 Statement of the problem

The problem of (0, 1) hyperbolic programming consists of minimizing the function

$$F(x_1, \ldots, x_n) = \frac{a_0 + \sum_{i=1}^{n} a_i x_i}{b_0 + \sum_{i=1}^{n} b_i x_i}, \tag{1}$$

$$x_i = \{0, 1\}, \tag{2}$$

$$\begin{aligned} & a_i \geq 0 \\ & b_i \geq 0 \qquad i = 1, 2, \ldots, n. \\ & a_i + b_i > 0 \end{aligned} \tag{3}$$

We shall refer in the future to problems (1), (2), and (3) as problem 1.

We notice that there is no loss of generality in assuming that the b_i $(i = 1, 2, \ldots, n)$ are positive. Indeed if one b_i is null, the $a_i > 0$(3) and the variable x_i must take the value 0. Thus in the following we shall assume $b_i > 0$.

The approach to the problem 1 is given in [1]. A vector X^* is called an optimal solution of the problem 1 if X^* minimized (1), i.e., the function $F(X)$ attains a global minimum at the point X^*. The function $F(X)$ attains a local minimum at the point $X^0 = (x_1^0, \ldots, x_n^0)$ if for any vector $X = (x_1, \ldots, x_n)$ such that $\sum_{i=1}^{n} |x_i - x_i^0| = 1$ the inequality $F(X^0) \leq F(X)$ holds. The notions of local and global minimum are related as follows:

THEOREM 1: Any local minimum of the function $F(X)$ is also a global minimum.

The proof of theorem 1 is given in [1], page 155.

1.2 The algorithm

Using this theorem we derive an algorithm which constructs a local (global) minimum by scanning just once the set of fractions $\{a_i/b_i\}$.

We now describe the algorithm for the problem 1.

ALGORITHM 1: Let N denote the set of integer $\{1, 2, \ldots, n\}$, and let I be a subset of N.

(1) Set $I = \phi$ and set the index $j = 0$

(2) $j = j + 1$: If $j = n + 1$ go to 7

(3) If $\dfrac{a_j}{b_j} > \dfrac{a_0 + \sum_{i \in I} a_i}{b_0 + \sum_{i \in I} b_i}$ go to 2

(4) $I = I \cup \{j\}$ and $\dfrac{a_0}{b_0} = \dfrac{a_0 + a_j}{b_0 + b_j}$

(5) Set $K = \{j \in I \mid a_0/b_0 < a_j/b_j\}$ if $K = \phi$ go to 2

(6) Set $\dfrac{a_0}{b_0} = \dfrac{a_0 - \sum_{j \in K} a_j}{b_0 - \sum_{j \in K} b_j}$ and $I = I - K$ and go to 5

(7) Set $x_i^* = 1$ $i \in I$

$x_i^* = 0$ $i \in N - I$

$F = a_0/b_0$

Stop—an optimal solution is reached.

We shall demonstrate that the vector X^* obtained with this algorithm is the optimal solution for the (0, 1) hyperbolic programming problem. Let us first illustrate the algorithm with the following example that is also presented in [1].

We minimize the function $F(x_1, \ldots, x_n)$ where the fractions a_i/b_i are given in Table I.

TABLE I

i	0	1	2	3	4	5	6	7	8	9	10	11
$\frac{a_i}{b_i}$	$\frac{3}{6}$	$\frac{2}{1}$	$\frac{4}{8}$	$\frac{1}{3}$	$\frac{2}{5}$	$\frac{9}{15}$	$\frac{6}{10}$	$\frac{12}{25}$	$\frac{8}{18}$	$\frac{2}{6}$	$\frac{3}{3}$	$\frac{3}{7}$

The different steps described in Algorithm I are as follows:

(1) $I=\phi$. $\quad j=0$

(2) $j=1$

(3) as $2/1=a_1/b_1>a_0/b_0=3/6$ $\quad$ go to step 2

(2) $j=2$

(3) as $4/8=a_2/b_2=a_0/b_0=3/6$ $\quad$ go to step 4

(4) $\dfrac{a_0}{b_0}=\dfrac{7}{14}$ $\quad$ and $\quad I=\{2\}$

(5) $K=\phi$ $\quad$ and go to step 2

(2) $j=3$

(3) as $1/3=a_3/b_3<a_0/b_0=7/14$. $\quad$ go to step 4

(4) $a_0/b_0=8/17$ $\quad$ and $\quad I=\{2, 3\}$

(5) $K=\{2\}$

(6) $a_0/b_n=4/9$. $\quad$ go to 2

(2) $j=4$

(3) as $2/5<4/9$. $\quad$ go to 4

(4) $I=\{3, 4\}$ $\quad$ and $\quad a_0/b_0=6/14$

(5) $K=\phi$. $\quad$ go to 2.

Steps (2) and (3) are performed several times to eliminate the fractions 9/15, 6/10, 12/25, and 8/18. Then the fraction 2/6 is retained and $I=\{3, 4, 9\}$. As $K=\phi$ the set I remains unchanged and the procedure terminates by eliminating the last two fractions in the list, 3/3 and 3/7.

At the end $I=\{3, 4, 9\}$ and $a_0/b_0=8/20$.

The global solution is therefore

$$x_i^*=1 \qquad i\epsilon\{3, 4, 9\}=I$$

$$x_i^*=0 \qquad i\epsilon N-I.$$

hence $X^*=00110000100$.

The value of $F(X^*)$ equals $a_0/b_0=8/20$.

THEOREM 2: The solution X^* obtained by the algorithm is the optimal solution of the hyperbolic programming problem.

Before proving this theorem we want to state a useful result on the addition of fractions.

LEMMA 1: Consider two fractions a_1/b_1 and a_2/b_2, where $a_i > 0$, $b_i > 0$, $i = 1, 2$. Then we have

$$\min\left[\frac{a_1}{b_1}, \frac{a_2}{b_2}\right] < \frac{a_1 + a_2}{b_1 + b_2} < \max\left[\frac{a_1}{b_1}, \frac{a_2}{b_2}\right],$$

whenever $(a_1 b_2 - a_2 b_1) \neq 0$. The proof is immediate.

PROOF OF THEOREM 2: It is sufficient by Theorem 1 to prove that the vector X^* obtained by the algorithm is a local minimum of the problem.

It is easily seen that the elements of I obtained at the end of the algorithm are indices of fractions a_j/b_j that have the properties:

$$\frac{a_j}{b_j} \leq \frac{a_0 + \sum_{i \in I} a_i}{b_0 + \sum_{i \in I} b_i} = F(X^*) \qquad \text{for all} \qquad j \in I$$

$$\frac{a_j}{b_j} > \frac{a_0 + \sum_{i \in I} a_i}{b_0 + \sum_{i \in I} b_i} = F(X^*) \qquad \text{for all} \qquad j \in N - I.$$

We shall use this property to show that if any one x_i^* is replaced by $\bar{x}_i^*$ in $F(X^*)$ the value of the function increases.

Indeed by Lemma 1

$$\frac{a_0 + \sum_{i \in I} a_i + a_j}{b_0 + \sum_{i \in I} b_i + b_j} > \frac{a_0 + \sum_{i \in I} a_i}{b_0 + \sum_{i \in I} b_i} = F(X^*) \qquad \text{for all} \qquad j \in N - I$$

and

$$F(X^*) = \frac{a_0 + \sum_{i \in I} a_i - a_j + a_j}{b_0 + \sum_{i \in I} b_i - b_j + b_j} \leq \max\left[\frac{a_j}{b_j}, \frac{a_0 + \sum_{i \in I} a_i - a_j}{b_0 + \sum_{i \in I} b_i - b_j}\right] \qquad \text{for all} \qquad j \in I.$$

As $\dfrac{a_j}{b_j} \leq F(x^*)$ for $j \in I$ we must have that

$$F(x^*) \leq \frac{a_0 + \sum_{i \in I} a_i - a_j}{b_0 + \sum_{i \in I} b_i - b_j} \qquad \text{for all} \qquad j \in I.$$

This proves that X^* is a local solution and by Theorem 1 a global solution to the hyperbolic programming problem.

We note that the solution of problem 1 need not necessarily be unique, however, the algorithm given here produces the optimal solution which has the largest possible number of non null variables.

Further we remark the following.

COROLLARY: Let $\dfrac{a_{i_1}}{b_{i_1}} \leq \dfrac{a_{i_2}}{b_{i_2}} \leq \ldots \leq \dfrac{a_{i_n}}{b_{i_n}}$ be the n fractions a_i/b_i ordered in an nondecreasing order.

Let k, $1 \le k \le n$, be an integer such that

$$\frac{a_0 + \sum_{j=1}^{k} a_{i_j}}{b_0 + \sum_{j=1}^{k} b_{i_j}} \le \frac{a_{i_s}}{b_{i_s}} \qquad \text{for } s > k,$$

$$\frac{a_0 + \sum_{j=1}^{k} a_{i_j}}{b_0 + \sum_{j=1}^{k} b_{i_j}} \ge \frac{a_{i_s}}{b_{i_s}} \qquad \text{for } s \le k,$$

then the vector X^*, defined by

$$x^*_{i_j} = 1 \qquad j \le k$$

and

$$x^*_{i_j} = 0 \qquad j > k$$

is the solution of problem 1.

1.3 Evaluation and Comparison of the Algorithm 1

The algorithm 1 described here and the one described in [1] page 153 that we denote as algorithm 1*, were both programmed in Fortran IV and utilised to find the optimal solution of the problem (1), (2), (3) when the fractions a_i/b_i, $i = 0, 1, \ldots, n$ are randomly generated (a_i and b_i are both random integers between 0 and 1000). The work was done on a CDC 6400 computer. For values of $n = 1000\ (1000)\ 10000$ this experiment was repeated 100 times and the central processing time for each algorithm to produce the solution was recorded. A summary of the results for both algorithms is given in Figure 1 where for each n we have plotted the average computing time (in sec) to attain a solution of a problem with n fractions.

One realizes that in the range studied, the average computing time for both algorithms increases almost linearly with n; the line corresponding to algorithm 1 has, however, a smaller slope than the one for algorithm 1*.

An interesting property of algorithm 1 is that the number of arithmetical or logical operations it

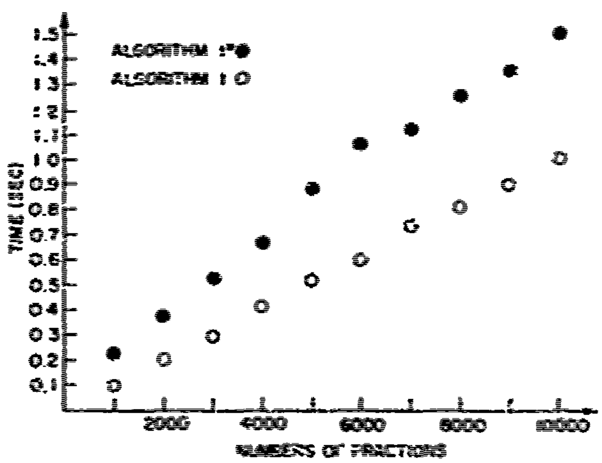


FIGURE 1.

needs to attain a solution is of the order of n^2. This can be proved easily by considering a problem where all the n fractions considered at the beginning are needed in the solution.

PART 2—THE HYPERBOLIC PROGRAMMING PROBLEM WITH CONSTRAINTS

2.1 Statement of the Problem

In the boolean algebra B_2 we define $x \leq y$ if and only if $xy = x$, $(x, y \in B_2)$. This relation coincides with the usual ordering of the set $\{0, 1\}$. In the space B_2^n consisting of the vectors $(x_1, x_2, \ldots, x_n)$ with $x_i \in B_2$ for each i, we define the following partial ordering $(x_1, x_2, \ldots, x_n) \leq (x_1', x_2', \ldots, x_n')$ if and only if $x_i \leq x_i'$ for each i. We define also $(x_1, x_2, \ldots, x_n) < (x_1', x_2', \ldots, x_n')$ if and only if $x_i \leq x_i'$ for each i and $x_j < x_j'$ for at least one j.

Consider now a function $f : B_2^n \to R$. We say that f is nondecreasing if, and only if, $f(x_1, x_2, \ldots, x_n) \leq f(x_1', x_2', \ldots, x_n')$, wherever $(x_1, x_2, \ldots, x_n) \leq (x_1', x_2', \ldots, x_n')$.

Examples of nondecreasing functions are

$$f(x_1, x_2, x_3) = 3x_1 + 0.5x_2 + 10x_3$$

or

$$f(x_1, x_2, x_3) = 3x_1x_2x_3 - 12\bar{x}_1 + x_2x_3.$$

The problem is to find a vector X^* which minimizes the function

$$F(x_1, x_2, \ldots, x_n) = \frac{a_0 + \sum_{i=1}^{n} a_i x_i}{b_0 + \sum_{i=1}^{n} b_i x_i}, \tag{4}$$

$$x_i = \{0, 1\}, \tag{5}$$

$$a_i \geq 0, \tag{6}$$

$$b_i > 0, \qquad i = 1, 2, \ldots, n, \qquad \text{and}$$

$$\begin{aligned} f_1(x_1, \ldots, x_n) &\geq F_1 \\ f_2(x_1, \ldots, x_n) &\geq F_2 \\ &\vdots \\ f_k(x_1, \ldots, x_n) &\geq F_k, \end{aligned} \tag{7}$$

where each function $f_i(X)$ is nondecreasing.

We shall refer to this problem as problem 2 and denote its solution by $X^* = (x_1^*, \ldots, x_n^*)$. We have immediately Lemma 2.

LEMMA 2: Problem 2 is feasible if and only if the vector $(1, 1, \ldots, 1)$ satisfies the constraints (7).

From now on we shall suppose that the above condition is always satisfied. The first thing one is tempted to do is to order the fractions in increasing order

$$\frac{a_{i_1}}{b_{i_1}} \leq \frac{a_{i_2}}{b_{i_2}} \leq \ldots \leq \frac{a_{i_n}}{b_{i_n}},$$

and to consider the variables $x_{i_1}, x_{i_2}, \ldots$ sequentially setting each 1 to 1 until the constraints are

satisfied. This does not give, in general, the optimal solution. Consider the following counterexample

$$\frac{a_0}{b_0}=\frac{1}{2},\ \frac{a_1}{b_1}=\frac{4}{5},\ \frac{a_2}{b_2}=\frac{1}{1}.$$

We want to minimize

$$\frac{a_0+a_1x_1+a_2x_2}{b_0+b_1x_1+b_1x_2}$$

where $a_1x_1+a_2x_2>0$.

It is easily seen that the solution if (0, 1) and not (1, 0).

THEOREM 3: If the vector $X^*=(x_1^*, \ldots, x_n^*)$ minimizes (4) with respect to the constraints (7), and if I is a set of integers such that

$$x_i^*=1 \qquad i\epsilon I\subseteq N$$

$$x_i=0 \qquad i\epsilon N-I,$$

then

$$F(X^*)=\frac{a_0+\sum_{i\epsilon I}a_i}{b_0+\sum_{i\epsilon I}b_i}\leq\frac{a_j}{b_j} \qquad j\epsilon N-I.$$

PROOF: Let us suppose the contrary, that is

$$\frac{a_0+\sum_{i\epsilon I}a_i}{b_0+\sum_{i\epsilon I}b_i}>\frac{a_j}{b_j} \qquad \text{for one} \qquad j\epsilon N-I.$$

Then by Lemma 1 we have

$$\frac{a_0+a_j+\sum_{i\epsilon I}a_i}{b_0+b_j+\sum_{i\epsilon I}b_j}<F(X^*).$$

Because the vector $(x_1^*, \ldots, \bar{x}_j^*, \ldots, x_n^*)\geq X^*$ it satisfies the constraints (7) and hence X^* is not the vector minimizing (4), a contradiction.

The solutions of the constrained and unconstrained problems are related as follows:

COROLLARY 3.1: Let us denote by X^0 the vector that minimizes (4) and has the smallest possible number of non null variables and denote by X^* a vector that minimizes (4) and satisfies the set of constraints (7). Then $X^0\leq X^*$. This follows directly from the last theorem.

COROLLARY 3.2: Let us denote by X^1 the vector which minimizes (4) and has the largest possible number of non null variables. If this vector does not satisfy the set of constraints (7) then $X^1<X^*$. This corollary states that if no solutions of problem I satisfy the constraints (2), X^* has as non null variables all the non null variables of X^1 and others in addition. This leads us to consider the following problem which is that of minimizing:

$$F(X)=\frac{a_0+\sum_{i=1}^{m}a_ix_i}{b_0+\sum_{i=1}^{m}b_ix_i}. \tag{8}$$

(9) $$a_0/b_0 < a_i/b_i \qquad i=1, 2, \ldots, m,$$

(10) $$a_0 \geq 0, \qquad b_0 > 0.$$

and subject to the new set of constraints deduced in an obvious way from the old ones

(11) $$\begin{gathered} g_1(x_1, \ldots, x_m) > G_1 \\ \vdots \\ g_k(x_1, \ldots, x_m) > G_k. \end{gathered}$$

This problem is referred as problem 3 and we denote by $(x_1^*, x_2^*, \ldots, x_m^*)$ a vector which minimizes (8) and satisfies the set of constraints (11), this with the smallest possible number of non-null variables. Let I be a subset of the set of integers $M=\{1, 2, \ldots, m\}$, where

$$\begin{gathered} x_i^*=1 \qquad \text{if } i\epsilon I \\ x_i^*=0 \text{ otherwise.} \end{gathered}$$

then we have Theorem 4.

THEOREM 4: The vector $(x_1^*, \ldots, x_m^*)$ and the set I described above must satisfy the two following properties.—

(12) $$\frac{a_0+\sum_{i\epsilon I} a_i}{b_0+\sum_{i\epsilon I} b_i} \leq \frac{a_j}{b_j} \qquad j\epsilon M-I$$

and

(13) If the integer $j\epsilon I$ is such that $a_j/b_j = \max_{i\epsilon I}(a_i/b_i)$

then $(x_1^*, \ldots, \bar{x}_j^*, \ldots, x_m)$ does not satisfy the constraints (11).

PROOF: (12) is a transcription of Theorem 3. We need only to prove that (13) holds.

By Lemma 1 and since $a_0/b_0 < a_j/b_j$, we have

$$F(X^*) \leq \frac{a_j}{b_j}, \text{ and}$$

thus $F(x_1^*, \ldots, \bar{x}_j^*, \ldots, x_m^*) \leq F(x_1^*, \ldots, x_m^*)$.

If (13) does not hold, i.e., if $(x_1^*, \ldots, \bar{x}_j^*, \ldots, x_m^*)$ satisfies the constraints (11), then we either have a better solution than the one given by X^* or at least one solution with a smaller number of non null variables. In both cases, a contradiction results.

We now describe an algorithm which gives a solution to the hyperbolic programming problem with the constraints given in (11).

We shall describe the information needed at each stage of the algorithm and the procedure to up date this information from one stage to the other.

ALGORITHM II: At the kth stage, the algorithm generates a vector $X^k=(x_1^k, x_2^k, \ldots, x^k)$ which minimizes (8) in the class of vectors that satisfy (11) and that have at most $k-1$ non null variables. It

also generates a set C_k of vectors defined as follows:

$$X=(x_1,\ldots,x_m)\in C_k \text{ if, and only if,}$$

$$\begin{cases} -\sum_{i=1}^{m} x_i = k-1 \\ -X \text{ does not satisfy (11)} \\ -F(X) \leq \frac{a_j}{b_j} \text{ for all } j \text{ such that } x_j=0 \\ -F(X) < F(X^k). \end{cases}$$

To advance from stage k to stage $k+1$ one proceeds as follows:

$$-\text{Set } X^{k+1}=X^k$$

For each vector $X'\in C_k$ from the class $L(X')$ of vectors $X=(x_1,\ldots,x_m)$ with the following properties:

$$\begin{cases} -X' < X \\ -\sum_{i=1}^{m} x_i = k \\ -F(X) \leq \frac{a_j}{b_j} \text{ for all } j \text{ such that } x_j=0 \\ -F(X) < F(X^{k+1}). \end{cases}$$

Let us define

$C_{k+1} = \bigcup_{X'\in C_k} \{X|X\in L(X') \text{ and } X \text{ does not satisfy (11)}\}$

If $\min_{X\in C^*_{k+1}} F(X) < F(X^{k+1})$ where

$C^*_{k+1} = \bigcup_{X'\in C_k} \{X|X\in L(X') \text{ and } X \text{ satisfies (11)}\}$

set $X^{k+1}=X^*$ where X^* is a vector in C^*_{k+1} such that $F(X^*) = \min_{X\in C^*_{k+1}} F(X)$.

If $C_{k+1}=\phi$. Set $X^*=X^{k+1}$ and stop. An optimal solution is reached.
If $C_{k+1}\neq\phi$, go to step $k+2$.
At stage 1 we let $X^1=(1,1,\ldots,1)$ and $C_1=(0,0,\ldots,0)$.
This terminates the description of the algorithm.

THEOREM 5: If the procedure stops at step $k+1$ then X^{k+1} is the vector which minimizes (8) and satisfies (11). This with the smallest possible number of non-null variables.

PROOF: Let X^* be the vector which minimizes (8) subject to (5) and has the smallest possible number, say r, on non-null variables.

Let j be the integer such that

$x_j^*\neq 0$ and $\frac{a_j}{b_j}\geq\frac{a_i}{b_i}$ for all i such that $x_i^*\neq 0$.

By Theorem 4, $(x_1^*,\ldots,\bar{x}_j^*,\ldots,x_m^*)$ does not satisfy (5), also $(x_1^*,\ldots,\bar{x}_j^*,\ldots,x_m^*)$ has $r-1$ non-

null variables and we know that $F(x_1^*, \ldots, \bar{x}_j^*, \ldots, x_m^*) \leq F(X^*) \leq F(x^r)$ where x^r is any vector that has at most $r-1$ non-null variables and satisfies (11).
Also it is easily seen that

$$F(x_1^*, \ldots, \bar{x}_j^*, \ldots, x_m^*) \leq \frac{a_k}{b_k} \text{ for all } k \text{ such that } x_k^*=0 \text{ and } k \neq j.$$

Hence the vector $(x_1^*, \ldots, \bar{x}_j^*, \ldots, x_m^*)$ is in C_r and at stage $r+1$ the vector X^{r+1} will take the value X^*. If the procedure stops at stage $k+1$ we recognize that $X^{k+1}=X^*$ which proves the theorem.

The solution of problem 2 can be obtained by solving first problem 1 and obtaining X^u as the optimal solution of the unconstrained (0, 1) hyperbolic programming. If X^o does not satisfy the constraints (7) we reduce the original problem to the problem 3 and use algorithm II. An equivalent and faster procedure would be to start the algorithm II at stage $s+1$ where s is the number of non-null variables of X^0 and to set $C_{s+1}=X^o$ and $X^{s+1}=(1, 1, \ldots, 1)$.

EXAMPLE: This is the continuation of the first example. We want now to minimize the function

$$\frac{a_0+\sum^{11} a_i x_i}{b_0+\sum_1^{11} b_i x_i}$$

with the restriction

$$a_0+\sum_{i=1}^{11} a_i x_i > 18. \tag{12}$$

The solution to the unconstrained problem is

$$x^o=(0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0).$$

This vector does not satisfy the constraint (12). From Corollary 3.2 we know that the solution of the constrained problem has $x_3=x_4=x_9=1$.

The algorithm starts at stage 4, where C_4 and X^4 are defined as follows:

STAGE 4

$$X^4=(11111111111);\ F(X^4)=0.486$$
$$C_4=\{X^o\}=(00110000100)$$

STAGE 5:

$$L(X^o)=\begin{cases}(01110000100)\\(00110001100)\\(00110000101)\\(00110010100)\end{cases}$$

$$C_5=\begin{cases}(01110000100)=Y_1\\(00110001100)=Y_2\\(00110000101)=Y_3\end{cases}$$

$$X^5=(00110010100);\ F(X^5)=0.444$$

STAGE 6:

$$L(Y_1) = (01110000101)$$

$$L(Y_2) = (00110001101)$$

$$L(Y_3) = \begin{bmatrix} (01110000101) \\ (00110001101) \\ (00110010101) \end{bmatrix}$$

$$C_6 = (01110000101) = Z$$

$$X^6 = (00110001101);\ F(X^6) = 0.422$$

STAGE 7:

$$L(Z) = \phi$$

$$C_7 = \phi$$

The solution is then $(00110001101) = X^*$ with $F(X^*) = 0.422$.

REFERENCE

[1] Hammer, P. L. and S. Rudeanu. *Boolean Methods in Operations Research and Related Areas* (Springer, New York, 1968).

APPLICATION OF THE GLM TECHNIQUE TO A PRODUCTION PLANNING PROBLEM*

J. P. Evans † and F. J. Gould ††

University of North Carolina at Chapel Hill

Abstract

Generalized Lagrange Multipliers (GLM) are used to develop an algorithm for a type of multiproduct single period production planning problem which involves discontinuities of the fixed charge variety. Several properties of the GLM technique are developed for this class of problems and from these properties an algorithm is obtained. The problem of resolving the gaps which are exposed by the GLM procedure is considered, and an example involving a quadratic cost function is explored in detail.

1. INTRODUCTION

This paper continues and extends the type of investigation reported in an earlier paper [4] dealing with applications of Generalized Lagrange Multipliers to a class of production planning problems which involve fixed charges in the form of setup times and costs. The method of Generalized Lagrange Multipliers was developed by Everett [5] for the general mathematical programming problem.

$$(P)\qquad \max_{x\in D} \quad f(x)$$

$$\text{s.t.} \qquad g_j(x) \le \bar{b}_j,\ j=1,\ldots,m,$$

where D is some subset of R^n on which f and g_j, $j=1,\ldots,m$, are real valued. The principal part of the GLM technique requires solution of the problem.

$$\max_{x\in D} \quad [F(x,\lambda)]$$

where F is the Lagrangian,

$$F(x,\lambda)=f(x)-\sum_{j=1}^{m}\lambda_j g_j(x),$$

and the multiplier values λ_j, are fixed and nonnegative. Everett proved that if x^* maximizes $F(x,\lambda)$ over the set D, then x^* is optimal in problem (P) modified by replacing the value of $\bar{b}_j$ with $g_j(x^*)$, $j=1,\ldots,m$. In order to solve (P) for a specific $\bar{b}$, it is easily shown to be sufficient [2], [6] to find a $\lambda^*\ge 0$, and an x^* which maximizes $F(x,\lambda^*)$ over D, such that $g_j(x^*)\le \bar{b}_j$, and $\lambda_j^*(g_j(x^*)-\bar{b}_j)=0$, $j=1,\ldots,m$.

In this paper we consider a single period, multiproduct production planning model, for which we develop a procedure for obtaining a set of multipliers, λ_j^*, for which under certain conditions the

*This work was supported by the Office of Naval Research, Contract No. N00014-67-A-0321-0003(NR047-095).

†Graduate School of Business Administration, University of North Carolina at Chapel Hill.

††Department of Statistics, University of North Carolina at Chapel Hill.

Preceding page blank

above optimality conditions are satisfied. The model is related to but different from the one investigated in [6]. A general algorithm is described and a specific case involving quadratic cost functions is explored in detail.

In the general model the objective is profit maximization, unit revenue is constant, and everything produced can be sold. A constant production rate is assumed, each product produced requires a setup time, and the total time available is limited. We assume the cost incurred in producing a product can be expressed as a function of only the total time (setup plus production) allocated to that product and that the cost of time allocated is given by a convex increasing function. This might be justified by an assumption that prices of aggregate inputs (including possibly labor, raw materials, facilities, etc.) are increasing.

For each product $i = 1, \ldots, K$, let

x_i = total production hours, excluding setup time.

R_i = revenue per production hour (> 0).

S_i = setup time (> 0).

$C_i(\xi)$ = a strictly convex function, defined for all ξ, nonnegative and increasing for $\xi \geq 0$: $C_i(\xi)$ is the cost of allocating a total of ξ hours to product i. We shall assume $C_i(0) = 0$, C_i is twice continuously differentiable, and $C_i'(x_i)$ unbounded.

$$\delta(\xi) = \begin{cases} 0 \text{ if } \xi = 0 \\ 1 \text{ if } \xi > 0. \end{cases}$$

w = a constant giving the total number of hours available.

The problem is then

$$\max \sum_{i=1}^{K} [R_i x_i - C_i(x_i + \delta(x_i)S_i)] \tag{1.1}$$

$$\text{s.t.} \sum_{i=1}^{K} [x_i + \delta(x_i)S_i] \leq w \tag{1.2}$$

$$x_i \geq 0, \; i = 1, \ldots, K.$$

Since this version of the model contains only one constraint we will need only one Lagrange multiplier, λ.

2. MAXIMIZING THE LAGRANGIAN

At this point we wish to employ a nondegeneracy assumption that for each i, if product i were the only product in the problem, then it would be profitable to produce that product up to some positive

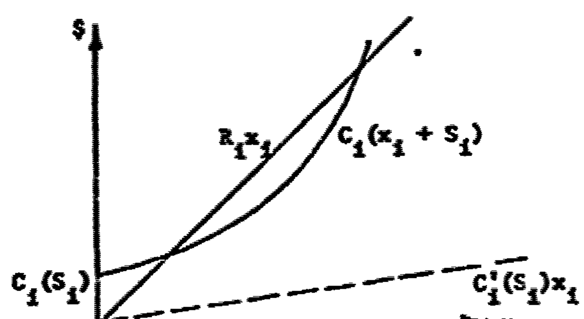


FIGURE 1.

level. That is, for at least one value of $x_i > 0$ we have

$$R_i x_i - C_i(x_i + S_i) > 0.$$

By the fact that C_i is a convex strictly increasing function with value zero at the origin, it follows that the above nondegeneracy assumption implies $R_i > C_i'(S_i)$, $i = 1, \ldots, K$. This situation is exhibited in Figure 1, from which is is apparent that if R_i were not greater than $C_i'(S_i)$, then $R_i x_i - C_i(x_i + S_i) < 0$ for each $x_i > 0$.*

For the problem under consideration the Lagrangian function is given by

$$F(x, \lambda) = \sum_{i=1}^{K} [(R_i - \lambda)x_i - C_i(x_i + \delta(x_i)S_i) - \lambda\delta(x_i)S_i].$$

The rules for maximizing this function are derived in the appendix. For a given $\lambda \geq 0$, let $x^*(\lambda)$ denote the production plans which maximize the Lagrangian. For $i = 1, \ldots, K$ †

$$x_i^*(\lambda) = \begin{cases} c_i(R_i - \lambda) - S_i, & \text{if } 0 \leq \lambda < \hat{\lambda}_i; \\ 0 \text{ or } c_i(R_i - \hat{\lambda}_i) - S_i, & \text{if } \lambda = \hat{\lambda}_i \\ 0 & \text{if } \lambda > \hat{\lambda}_i, \end{cases} \tag{2.1}$$

where $\hat{\lambda}_i$ is the unique zero of the function

$$H_i(\lambda) = (R_i - \lambda)[c_i(R_i - \lambda) - S_i] - C_i(c_i(R_i - \lambda)) - \lambda S_i \tag{2.2}$$

on the open interval $(0, R_i - C_i'(S_i))$. For the time being there is no need to distinguish between the possible alternative values of $x_i^*(\lambda)$ for the case $\lambda = \hat{\lambda}_i$.

3. CONSTRAINT AND OBJECTIVE FUNCTION PROPERTIES

In this section we consider the effect in the production time constraint of using a production plan from (2.1) which maximizes the Lagrangian. Define:

$$T(\lambda) = \sum_{i=1}^{K} (x_i^*(\lambda) + \delta(x_i^*(\lambda))S_i);$$

$T(\lambda)$ is then the aggregate time allocated to production and setup by the plan $x^*(\lambda)$. Because of the existence of alternative optima in (2.1), $T(\lambda)$ is multiple-valued for some values of λ.‡ In this case the symbol $T(\lambda)$ will be used to represent any one of the possible values. Discrimination among these multiple values will be provided as the developments require.

---

* The necessity of $R_i > C_i'(S_i)$ is demonstrated analytically as follows. Since $R_i x_i - C_i(x_i + S_i)$ is a concave function, we have for each $x_i > 0$, $R_i x_i - C_i(x_i + S_i) \leq -C_i(S_i) + (R_i - C_i'(S_i))x_i$. If $R_i \leq C_i'(S_i)$ the right hand side of this inequality is negative for each positive x_i.

† In (2.1) $c_i(\cdot)$ denotes the inverse function of $C_i'(\cdot)$. That is $C_i'(\xi) = z \Leftrightarrow c_i(z) = \xi$. The assumptions on $C_i(\cdot)$ guarantee that this inverse function is defined on the set of real numbers $\{z : z \geq C_i'(0)\}$.

‡ These values of λ are precisely the λ values $\hat{\lambda}_i$, $i = 1, \ldots, K$, mentioned above in connection with (2.1) and (2.2). As we shall see in the example in section 4, these values can be computed at the outset of our analysis of the problem.

PROPERTY 1: If $T(\lambda) > 0$ and $\lambda' > \lambda$, then $T(\lambda') < T(\lambda)$.

PROOF: If $T(\lambda) > 0$, then $x_i^*(\lambda) > 0$ for at least one i. If $x_i^*(\lambda) > 0$, (2.1) implies $\lambda \leq \hat{\lambda}_i$, and $x_i^*(\lambda) = c_i(R_i - \lambda) - S_i$.

a) If $\lambda < \lambda' \leq \hat{\lambda}_i$, then $x_i^*(\lambda') < x_i^*(\lambda)$ either because $x_i^*(\lambda_i')$ drops to zero or because $c_i(\cdot)$ is an increasing function of its argument.

b) If $\hat{\lambda}_i < \lambda'$, $x_i^*(\lambda') = 0$.

PROPERTY 2: For λ sufficiently large, $T(\lambda) = 0$.

PROOF: If $\lambda > \max\,[\hat{\lambda}_i : i = 1, \ldots, K]$, $x_i^*(\lambda) = 0$, $i = 1, \ldots, K$.

PROPERTY 3: Let $\lambda > 0$ be fixed and let $\pi(j)$ be a permutation of the integers $1, \ldots, K$, such that $\hat{\lambda}_{\pi(1)} \leq \hat{\lambda}_{\pi(2)} \leq \ldots \leq \hat{\lambda}_{\pi(K)}$. For convenience it will be assumed that these inequalities can be taken as strict. Also define $\hat{\lambda}_{\pi(0)} = 0$. Then

1) For $\hat{\lambda}_{\pi(l-1)} < \lambda < \hat{\lambda}_{\pi(l)}$, $l = 1, \ldots, K$, $T(\cdot)$ is a single valued continuous, strictly decreasing function.

2) At $\lambda = \hat{\lambda}_{\pi(l)}$, $l = 1, \ldots, K$, $T(\cdot)$ has a discrete jump of the amount $c_{\pi(l)}(R_{\pi(l)} - \hat{\lambda}_{\pi(l)})$.

3) For $\lambda > \hat{\lambda}_{\pi(K)}$, $T(\lambda) = 0$.

PROOF: Conclusion 3) is a restatement of Property 2. Now consider part 1), and let λ be any point in one of the open intervals $(\hat{\lambda}_{\pi(l-1)}, \hat{\lambda}_{\pi(l)})$, $l = 1, \ldots, K$. Since $\lambda < \hat{\lambda}_{\pi(l)}$ it follows from (2.1) that for for $j = l, \ldots, K$,

$$x_{\pi(j)}^*(\lambda) = c_{\pi(j)}(R_{\pi(j)} - \lambda) - S_{\pi(j)};$$

whereas for $j = 1, \ldots, l-1$, $x_{\pi(j)}^*(\lambda) = 0$. Thus $T(\lambda)$ is single valued, and

$$T(\lambda) = \sum_{j=l}^{K} (x_{\pi(j)}^*(\lambda) + S_{\pi(j)}) = \sum_{j=l}^{K} c_{\pi(j)}(R_{\pi(j)} - \lambda).$$

As λ varies on the open interval from $\hat{\lambda}_{\pi(l-1)}$ to $\hat{\lambda}_{\pi(l)}$, $T(\lambda)$ is strictly decreasing and continuous, because each of the functions $c_{\pi(j)}$ is increasing and continuous in its argument. This establishes conclusion 1).

Now consider the case $\lambda = \hat{\lambda}_{\pi(l)}$ for some $l \in \{1, \ldots, K\}$. By (2.1), either $x_{\pi(l)}^*(\lambda) = 0$ or $x_{\pi(l)}^*(\lambda) = c_{\pi(l)}(R_{\pi(l)} - \hat{\lambda}_{\pi(l)}) - S_{\pi(l)}$; however, we need not be concerned at this point, with the precise value of T at $\lambda = \hat{\lambda}_{\pi(l)}$. Rather, we focus on limits of T as λ approaches $\hat{\lambda}_{\pi(l)}$ from either side. In particular, as λ approaches $\hat{\lambda}_{\pi(l)}$ from below we have

$$\lim_{\lambda \nearrow \hat{\lambda}_{\pi(l)}} x_{\pi(l)}^*(\lambda) = c_{\pi(l)}(R_{\pi(l)} - \hat{\lambda}_{\pi(l)}) - S_{\pi(l)},$$

and thus

$$\lim_{\lambda \nearrow \hat{\lambda}_{\pi(l)}} T(\lambda) = \sum_{j=l}^{K} c_{\pi(j)}(R_{\pi(j)} - \hat{\lambda}_{\pi(l)}); \tag{3.1}$$

on the other hand as λ approaches $\hat{\lambda}_{\pi(l)}$ from above we have $x_{\pi(l)}^*(\lambda) = 0$ for $\lambda > \hat{\lambda}_{\pi(l)}$, and

$$\lim_{\lambda \searrow \hat{\lambda}_{\pi(l)}} T(\lambda) = \sum_{j=l+1}^{K} c_{\pi(j)}(R_{\pi(j)} - \hat{\lambda}_{\pi(l)}). \tag{3.2}$$

where $\nearrow(\searrow)$ denotes monotonic convergence from below (above). Equations (3.1) and (3.2) imply that $T(\cdot)$ has a discrete jump at $\hat{\lambda}_{\pi(l)}$ by the amount $c_{\pi(l)}(R_{\pi(l)} - \hat{\lambda}_{\pi(l)})$, completing the proof.

These properties of T are portrayed on Figure 2, where the T values at the points $\hat{\lambda}_i$ are unspecified.

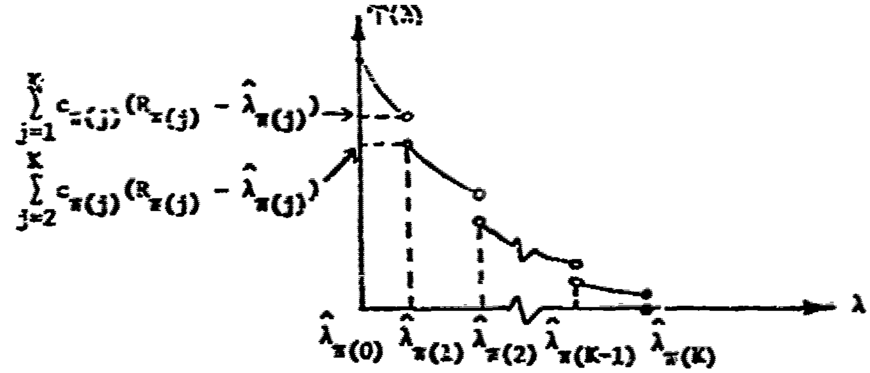


FIGURE 2.

On the basis of Property 3, the main portion of an algorithm for the original problem in (1.1) and (1.2) can be stated. Increase λ from zero to some value $\tilde{\lambda}$ such that $T(\tilde{\lambda}) = \kappa$ if possible. Then the associated production plan is an optimal solution. This follows from the properties of the Generalized Lagrange Multiplier technique proved by Everett [5]. If this procedure is not possible, then κ is said to be in a gap. This means there is an $l \in \{1, \ldots, K\}$, such that when $\lambda = \hat{\lambda}_{\pi(l)}$ each of the possible values of $T(\lambda)$ is either $< \kappa$ or $> \kappa$, but for each $\epsilon > 0$, $T(\lambda - \epsilon) > \kappa$ and $T(\lambda + \epsilon) < \kappa$. We can in this case construct lower and upper bounds on the optimum solution and thereby further refine the algorithm for the case when κ falls in a gap.

Suppose for the sake of convenience that the products are numbered in increasing order of the values $\hat{\lambda}_i$, i.e., $\hat{\lambda}_1 < \hat{\lambda}_2 < \ldots < \hat{\lambda}_K$. The following result will be of interest.

PROPERTY 4: Suppose $\lambda = \hat{\lambda}_l$ for some l, $1 \leq l \leq K$.

a) If $\kappa = \underline{\kappa} = \sum_{j=l}^{K} c_j(R_j - \hat{\lambda}_l)$, the optimal production plan is

$$x_j^*(\lambda) = \begin{cases} 0, & j = 1, \ldots, l-1 \\ c_j(R_j - \hat{\lambda}_l) - S_j, & j = l, \ldots, K. \end{cases}$$

b) If $\kappa = \tilde{\kappa} = \sum_{j=l+1}^{K} c_j(R_j - \hat{\lambda}_l)$, the optimal production plan is

$$x_j^*(\lambda) = \begin{cases} 0, & j = 1, \ldots, l \\ c_j(R_j - \hat{\lambda}_l) - S_j, & j = l+1, \ldots, K. \end{cases}$$

PROOF: The conclusions follow from the fact that both plans maximize the Lagrangian for $\lambda = \hat{\lambda}_l$. Thus if the resource availability is $\kappa = \tilde{\kappa}$, the plan in a) must be optimal, whereas if $\kappa = \underline{\kappa}$, the plan in b) is optimal, as a consequence of the properties of the GLM technique.

For the case $\underline{\kappa} < \kappa < \tilde{\kappa}$, we will use the results of Property 4 to develop lower and upper bounds on the optimal value of the objective function. Define $M(\kappa)$ = maximum profit attainable for resource level κ (see Figure 3). We have from Property 4 that

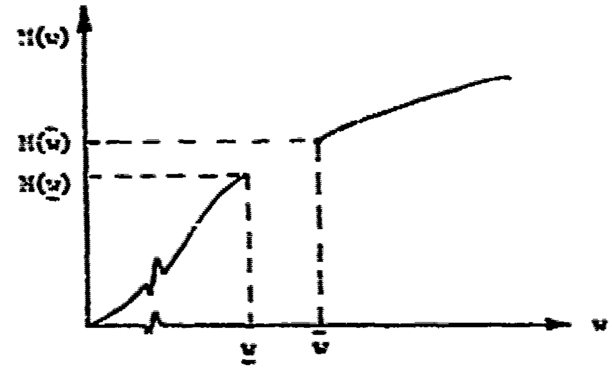


FIGURE 3.

$$\hat{w}-\underline{w}=c_t(R_t-\hat{\lambda}_t)$$

and

$$M(\hat{w})-M(\underline{w})=R_t(c_t(R_t-\hat{\lambda}_t)-S_t)-C_t(c_t(R_t-\hat{\lambda}_t)).$$

From results in [6] it follows that the function $M(\cdot)$ has a linear support with slope $\lambda=\hat{\lambda}_t$ at $\underline{w}$ and $\hat{w}$: that is

$$(3.3)\qquad \begin{cases} (w-\hat{w})\hat{\lambda}_t+M(\hat{w})\geq M(w), & \text{all } w \\ (w-w)\hat{\lambda}_t+M(\underline{w})\geq M(w), & \text{all } w. \end{cases}$$

By substituting $\underline{w}$ in the first of the above inequalities, and $\hat{w}$ in the second, it can be seen that

$$(3.4)\qquad \hat{\lambda}_t=\frac{M(\hat{w})-M(\underline{w})}{\hat{w}-\underline{w}},$$

and it follows that the two upper estimates in (3.3) are equally tight upper bounds on $M(w)$ for $w\in(\underline{w},\hat{w})$.†

It will now be useful to define the real profit function, for $i=1,\ldots,K$:

$$P_i(x_i)=\begin{cases} R_ix_i-C_i(x_i+S_i), & \text{if } x_i>0 \\ 0 & x_i=0. \end{cases}$$

Thus, P_i gives the contribution to total real profit accruing from the allocation of x_i time units (excluding setup) to the production of product i. It is clear that as x_i approaches zero from the right, P_i approaches the limit $-C_i(S_i)$. Also, for $x_i>0$, P_i is strictly concave with a unique global maximum at $\hat{x}_i$ which satisfies

$$R_i-C_i'(\hat{x}_i+S_i)=0, \qquad \text{or } c_i(R_i)-S_i=\hat{x}_i.$$

There exists such a value because C_i' is strictly increasing, unbounded, and $C_i'(S_i)<R_i$ by the non-

† In [1] a relation similar to (3.4) is used in constructing a multiplier search. In the current situation the search has been completed in that we have identified resource levels which bound a gap.

degeneracy assumption of section 2. Also note that $P_i'(x_i) > 0$ for $0 < x_i < \bar{x}_i$ and hence P_i is increasing in the range $(0, \bar{x}_i)$ with a unique zero in this interval, say at x_i^0. Then for product i to make a positive contribution to total profit, more than x_i^0 hours must be allocated to actual production (excluding setup) of product i. We also note that in the case where $x_i^*(\hat{\lambda}_i) = c_i(R_i - \hat{\lambda}_i) - S_i$, we have $x_i^*(\hat{\lambda}_i) < \bar{x}_i$ and

$$P_i(x_i^*(\hat{\lambda}_i)) = R_i x_i^*(\hat{\lambda}_i) - C_i(x_i^*(\hat{\lambda}_i) + S_i) > (R_i - \hat{\lambda}_i)x_i^*(\hat{\lambda}_i) - C_i(x_i^*(\hat{\lambda}_i) + S_i) - \hat{\lambda}_i S_i = G_i(x_i^*(\hat{\lambda}_i), \hat{\lambda}_i) = 0.\dagger$$

Hence,
$$x_i^0 < x_i^*(\hat{\lambda}_i) < \bar{x}_i.$$

Now suppose w is in a gap which occurs at $\hat{\lambda}_l$, and suppose $\underline{w} + x_l^0 + S_l < w < \bar{w}$. This means that there is enough time remaining (i.e., $w - \underline{w}$) so that positive profit could be made from the production of product l. In this case, if we set $x_l = w - S_l - \underline{w}$, we can increase the profit from $M(\underline{w})$ to $M(\underline{w}) + P_l(x_l)$ and still remain feasible. Note that $P(x_l) > 0$ because

$$x_l^0 < x_l < c_l(R_l - \hat{\lambda}_l) - S_l < \bar{x}_l. \tag{3.5}$$

Thus, for $w \in (\underline{w}, \bar{w})$, the gap occurring at $\hat{\lambda}_l$, for $\underline{w} + x_l^0 + S_l < w < \bar{w}$, and for the production plan,

$$x_i' = \begin{cases} 0, & i = 1, \ldots, l-1 \\ w - S_l - \underline{w}, & i = l \\ c_i(R_i - \hat{\lambda}_l) - S_i, & i = l+1, \ldots, K, \end{cases} \tag{3.6}$$

the number $M(\underline{w}) + P_l(x_l')$ is a lower bound on the optimal objective value, $M(w)$. The following result shows that (3.6) is not optimal.

PROPERTY 5: Suppose w is in a gap which occurs at $\hat{\lambda}_l$ and suppose $\underline{w} + x_l^0 + S_l < w < \bar{w}$, for some fixed $l < K$. Then

$$M(w) > M(\underline{w}) + P_l(w - S_l - \underline{w}).$$

PROOF: Employing (3.6), we have

$$P_i'(x_i') = R_i - C_i'(x_i' + S_i) = \hat{\lambda}_l > 0, \; i = l+1, \ldots, K, \tag{3.7}$$

and for $i = l$,

$$P_l'(x_l') = R_l - C_l'(x_l' + S_l) > R_l - C_l'(c_l(R_l - \hat{\lambda}_l)) = \hat{\lambda}_l > 0, \tag{3.8}$$

where we employ the above relations, (3.5) and the monotonicity of C_l'. Now consider the production plan

$$\tilde{x}_i = \begin{cases} 0 & i = 1, \ldots, l-1 \\ w - S_l - \underline{w} + (K-l)\epsilon, & i = l \\ c_i(R_i - \hat{\lambda}_l) - S_i - \epsilon, & i = l+1, \ldots, K. \end{cases} \tag{3.9}$$

---

†The function $G_i(\cdot, \cdot)$ and the equality employed here are developed in Equation (A.2) of the appendix.

where $\epsilon > 0$. This plan utilizes precisely the quantity u of the resource. Then (3.7) and (3.8) imply that for sufficiently small $\epsilon > 0$, the profit of the new plan will exceed the profit associated with the plan in (3.6), i.e.,

$$\sum_{i=I}^{K} P_i(\bar{x}_i) > M(\underline{u}) + P_I(u - S_I - \underline{u}).$$

For the case considered in Property 5, a lower bound on $M(u)$ that is strictly larger than $M(\underline{u}) + P_I(u - S_I - \underline{u})$ can be constructed by considering the auxiliary problem.

$$(\hat{A})$$

(3.10)
$$\max \sum_{i=I}^{K} [R_i x_i - C_i(x_i + S_i)]$$

(3.11)
$$\text{s.t. } \sum_{i=I}^{K} x_i \leqq u - \sum_{i=I}^{K} S_i$$

(3.12)
$$x_i \geqq 0, \ i = I, \ldots, K.$$

In problem $(\hat{A})$ the setup times of products I thru K have been subtracted from u and the associated costs for these setups will be charged in the objective function, regardless of the actual production plan. The Lagrangian for this auxiliary problem is

$$L(x, \lambda) = \sum_{i=I}^{K} L_i(x, \lambda) = \sum_{i=I}^{K} [(R_i - \lambda)x_i - C_i(x_i + S_i)].$$

Based on the analysis in the appendix it is easy to show that the plan which maximizes this Lagrangian is, for $i = I, \ldots, K$,

(3.13)
$$x_i(\lambda) = \begin{cases} c_i(R_i - \lambda) - S_i & \text{if } 0 \leq \lambda \leq R_i - C_i'(S_i) \\ 0 & \text{if } \lambda > R_i - C_i'(S_i). \end{cases}$$

Figure 4 indicates, as can be proved analytically, that $x_i(\lambda)$, $i = I, \ldots, K$, is continuous in λ for all $\lambda \geq 0$, and decreasing in λ for $0 \leq \lambda < R_i - C_i'(S_i)$.

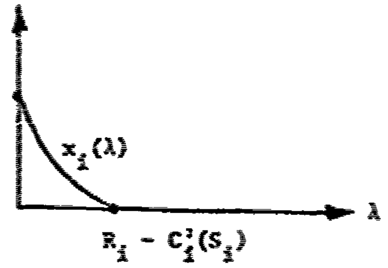


FIGURE 4.

The following result shows that the optimal solution to problem (A) yields a lower bound on $M(w)$ that is strictly better than the profit associated with the plan of (3.6).

PROPERTY 6:

a) There exists a $\lambda_0 > \hat{\lambda}_k$ such that for $x_i(\lambda_0)$, $i = 1, \ldots, \hat{k}$, as prescribed in (3.13), we have

$$\sum_{i=1}^{\hat{k}} x_i(\lambda_0) = w - \sum_{i=1}^{\hat{k}} S_i.$$

b) The plan $x_i(\lambda_0)$ is optimum for problem (A) and the associated value of the objective function exceeds $M(\underline{w}) + P_k(w - S_k - \underline{w})$.

PROOF: By the assumption that w is in a gap, we have

$$\sum_{i=1}^{\hat{k}} x_i(R_i - \hat{\lambda}_k) = \bar{w} - \sum_{i=1}^{\hat{k}} S_i > w - \sum_{i=1}^{\hat{k}} S_i. \tag{3.14}$$

Also we have that $\hat{\lambda}_i < \hat{\lambda}_{i+1} < \ldots < \hat{\lambda}_k$, by assumption, and for each i, $\hat{\lambda}_i < R_i - C_i'(S_i)$. Thus for $\lambda = \max\{R_i - C_i'(S_i) | i = 1, \ldots, \hat{k}\}$, it follows from (3.13) that the plan that maximizes the Lagrangian has $x_i(\lambda) = 0$, $i = 1, \ldots, \hat{k}$, hence $\sum_{i=1}^{\hat{k}} x_i(\lambda) = 0$. But since $x_i(\lambda)$ in (3.13) is continuous in λ for each i, so is the sum; therefore it follows that there is a value λ_0 such that

$$\sum_{i=1}^{\hat{k}} x_i(\lambda_0) = w - \sum_{i=1}^{\hat{k}} S_i. \tag{3.15}$$

It remains to show that $\lambda_0 > \hat{\lambda}_k$. Suppose $\lambda_0 \leq \hat{\lambda}_k$. Then $\lambda \leq \hat{\lambda}_i$, $i = 1, \ldots, \hat{k}$; but in this range $x_i(\lambda)$ is decreasing in λ. Thus $\lambda_0 \leq \hat{\lambda}_k$ together with (3.14) implies

$$\sum_{i=1}^{\hat{k}} x_i(\lambda_0) \geq \sum_{i=1}^{\hat{k}} x_i(\hat{\lambda}_k) > w - \sum_{i=1}^{\hat{k}} S_i.$$

But this contradicts the property of λ_0 established in (3.15), hence $\lambda_0 > \hat{\lambda}_k$ and a) is proved.

The optimality of the plan follows from the fact that we have exhibited a plan that maximizes the Lagrangian of (A) and exactly utilizes the available resource. To establish the second conclusion of b), we observe that the plan of (3.6) is feasible in (A). The plan (3.9) in the proof of Property 5 is also feasible in the auxiliary problem, and is strictly better than the plan of (3.6). Hence the optimal plan of the auxiliary problem is strictly better than the plan of (3.6).

GAP DISCUSSION

The preceding discussion does not establish that the optimal solution to problem (A) will be optimal for the original problem when w falls in a gap. There is, however, reason to believe that problem (A) will yield a good lower bound for $M(w)$ in gap situations. Since for $w = \bar{w}$ we know it is optimal to produce each product $i = 1, \ldots, \hat{k}$, we can investigate the gap between $\underline{w}$ and $\bar{w}$ by proceeding as if the problem possesses a kind of stability (see [3]) for w near $\bar{w}$. This strategy is supported by the fact that as w approaches $\bar{w}$ from below, it can easily be shown that the λ_0 of Property 6 approaches $\hat{\lambda}_k$.

A more complete attempt to resolve gaps is possible; however, it would, in general, require combinatorial techniques which have so far been avoided by the use of Generalized Lagrange Multipliers. The fact that a solution via GLM methods cannot be guaranteed for each w is offset by the important fact that a great deal of parametric information is readily available about the influence of various values of w on the optimum value of the objective function. In particular, let us refer to Figure 2. In the first step of optimality analysis, the analyst should compute the K values of $\hat{\lambda}_i$. Recall that these are the points at which the T function is double valued and for each i the two T values specify a gap in the right-hand side. By computing the two T values associated with each $\hat{\lambda}_i$ one can immediately identify those values of w which fall in gaps. Also, the optimal product mix can be determined without further computation for any value of w not in a gap. For such values an optimal plan can be computed by the algorithm below. For those values of w which do lie in a gap, it is often the case that in actual problems there is some latitude in the specification of the right-hand side. In this case it may therefore be possible for the analyst (again with reference to Figure 2) to specify an acceptable value of the right-hand side which does not lie in a gap, and hence for which an optimal plan can be computed.

We can summarize our proposed algorithm as follows:

1) Starting with $\lambda=0$ increase λ until we reach $\lambda^*=\inf\{\lambda : T(\lambda)\leq w\}$.

2) If $T(\lambda^*)=w$, stop; we have an optimal solution. Otherwise continue.

3) $T(\lambda^*)<w$. To improve the solution, determine the product, l, for which $\lambda^*=\hat{\lambda}_l$, and compute $\bar{w}$. $\underline{w}$ as defined in Property 4.

a) If $w<\underline{w}+x_l^0+S_l$, stop; use the plan for $w=\underline{w}$ as an approximate solution.

b) If $w\geq \underline{w}+x_l^0+S_l$, solve the associated auxiliary problem, (3.10), (3.11), (3.12) by finding the λ_A of Property 6. Use the resulting plan as an approximation.

4. AN EXAMPLE—THE QUADRATIC CASE

In this section we wish to apply the preceding developments to a small example in which the cost functions $C_i(\cdot)$ are all quadratic and of the form

$$C_i(x_i+S_i)=\alpha_i(x_i+S_i)+\frac{\beta_i}{2}(x_i+S_i)^2,$$

where $\alpha_i, \beta_i>0$. We will demonstrate the calculation of the critical values $\hat{\lambda}_i$ and the use of the algorithm to obtain production plans for a range of resource values. Consider products with the following data:

i	S_i	R_i	α_i	β_i
1	5	15	1	1
2	8	40	1	2

For the quadratic case, we have

$$C_i'(x_i+S_i)=\alpha_i+\beta_i(x_i+S_i)$$

thus

$$c_i(y)=\frac{y-\alpha_i}{\beta_i}-S_i.$$

Now we consider the determination of the critical numbers $\hat{\lambda}_i$. We have from Equation (A.2) of the appendix

$$G_i(x_i, \lambda)=(R_i-\lambda)x_i-\alpha_i(x_i+S_i)-\frac{\beta_i}{2}(x_i+S_i)^2-\lambda S_i.$$

By definition, $\hat{\lambda}_i$ is the value of λ such that G_i has a maximum value of 0, which implies that the equation $G_i(x_i, \hat{\lambda}_i)=0$ has exactly one real root. The value of λ for which this occurs can be determined by setting the discriminant of this quadratic equation to zero and solving for λ. Thus we seek a solution to the following equation in λ:

$$(4.1)\qquad (R_i-\alpha_i-\beta_iS_i-\lambda)^2-2\beta_i\left((\alpha_i+\lambda)S_i+\frac{\beta_i}{2}S_i^2\right)=0.$$

The roots are

$$\lambda=R_i-\alpha_i\pm\sqrt{2R_i\beta_iS_i}.$$

It is easy to show that

$$(4.2)\qquad \hat{\lambda}_i=R_i-\alpha_i-\sqrt{2R_i\beta_iS_i}.$$

and that if $R_i-C_i'(S_i)=R_i-(\alpha_i+\beta_iS_i)>0$, then $\hat{\lambda}_i>0$; see the lemma in the appendix. For the data tabulated above, we get the following:

i	$\hat{\lambda}_i$	x_i^0
1	1.75	2.22
2	3.2	3.75

Also included in the above table are the minimum profitable production quantities x_i^0; these quantities are obtained by solving the equations

$$P_i(x_i)=R_ix_i-\alpha_i(x_i+S_i)-\frac{\beta_i}{2}(x_i+S_i)^2=0$$

for each i: the roots are

$$x_i=\frac{1}{\beta_i}\left(R_i-\alpha_i-\beta_iS_i\pm\sqrt{(R_i-\alpha_i)^2-2R_i\beta_iS_i}\right).$$

and x_i^0 is the smaller of the two roots.

With the values of $\hat{\lambda}_i$ determined, we consider now the problem of obtaining maximum profit for a given availability of time, w.

$0<\lambda<\hat{\lambda}_1=1.75$:

$$x_1^*=9-\lambda; \quad x_2^*=11.5-.5\lambda \tag{4.3}$$

Thus $T(\lambda)=33.5-1.5\lambda$ for $0<\lambda<1.75$. Hence at $\lambda=\hat{\lambda}_1$, we have $\bar{w}=30.875$, and $\underline{w}=18.625$, where $\underline{w}$ is computed by setting $x_1=0$ and $x_2=11.5-.5\hat{\lambda}_1$. So for $30.875<w<33.5$, the optimal solution is obtained by finding λ, $0<\lambda<1.75$, such that the x_i^* from (4.3) satisfy

$$x_1^*+x_2^*+13=w.$$

$1.75<\lambda<3.25$:

$$x_1^*=0; \; x_2^*=11.5-.5\lambda \tag{4.4}$$

Thus $T(\lambda)=19.5-.5\lambda$ for $1.75<\lambda<3.25$; at $\lambda=\hat{\lambda}_2$ we have $\bar{w}=17.875$ and $\underline{w}=0$. Hence for $17.875<w<18.625$, the optimal solution is obtained by finding λ, $1.75<\lambda<3.25$, such that x_i^* from (4.4) satisfy $x_2^*+8=w$.

Now consider the gap in $T(\lambda)$ which occurs at $\lambda=1.75$. We can compute two points of the function $M(w)$, namely

$$M(\bar{w})=M(30.875)=80.98$$

$$M(\underline{w})=M(18.625)=59.48.$$

Now, using the results in Equations (3.3) and (3.4), we construct the following upper bound on $M(w)$ for $18.625<w<30.875$:

$$M(w)\leqq 59.48+1.75(w-18.625).$$

We know that the minimum profitable production of product 1 requires 7.22 (production plus setup) time units, thus suppose $25.845<w<30.875$; for definiteness assume $w=28.625$. Thus using (3.4) and the discussion following (3.6) we have

$$74.48\leqq M(28.625)\leqq 76.98. \tag{4.5}$$

Now by employing the results of Property 6 and the auxiliary problem in (3.10), (3.11), and (3.12) we can refine the lower bound. Solving the auxiliary problem via Equation (3.13) yields the following results:

$$\lambda=3.25; \; x_1^*=5.75; \; x_2^*=9.875$$

$$P(x_1)+P(x_2)=75.33.$$

Clearly the profit exceeds out lower bound in (4.5); in fact it can be shown that this production plan is optimal for $w=28.625$.

Finally let us briefly consider the gap in $T(\lambda)$ which occurs at $\lambda=3.25$. We have $T(3.25)=17.875$, whereas the minimum profitable production of product 2 requires only 1[illegible].75 total hours. For

$11.75 < w < 17.875$, the approach suggested in (3.13) would yield a solution such that $x_2 = w - 8$. For $w < 11.75$ the discussion of section 3 suggests no detailed remedy; however for this example it is clear that product 1 can be produced profitably provided only that $w > 7.22$.

5. CONCLUSION

In the foregoing discussion we have developed and demonstrated an application of the Generalized Lagrange Multiplier technique to a certain type of production planning problem. Particular attention has been devoted to the issue of resolving gaps in the function $T(\lambda)$. No complete solution was provided for these situations; however, useful bounds have been constructed and a procedure for refining these bounds was proposed. The primary goal of this paper and its companion [2] has been to exploit the analytic (i.e., non-combinatorial) nature of algorithms based on the Generalized Lagrange Multiplier technique.

An important property of the model in (1.1) and (1.2) is the fact that the critical values of the Lagrange Multiplier, the $\hat{\lambda}_i$, $i = 1, \ldots, K$, can, in principle, be calculated directly.* In fact in the quadratic case, a simple closed form expression was provided for $\hat{\lambda}_i$ in Equation (4.2). The method for exploiting this information was discussed in detail in Section 4. Furthermore, for the quadratic case, since $C_i'(\cdot)$ is linear, its inverse $c_i(\cdot)$ is likewise linear. This fact can be exploited in planning computations for the quadratic case.

Several possible extensions of this model and the algorithm are readily apparent. In particular a collection of one or more material constraints might be added; the current model assumes that any time-feasible model is also material-feasible. Other extensions include overtime and/or subcontracting alternatives. The model in [4] is of the latter type.

Appendix

MAXIMIZING THE LAGRANGIAN

The Lagrangian, $F(x, \lambda)$ is given by $\sum_{i=1}^{k} F_i(x_i, \lambda)$, where $F_i : R^+ \times R^+ \to R$, R^+ denoting the non-negative reals, and

$$F_i(x_i, \lambda) = (R_i - \lambda)x_i - C_i(x_i + \delta(x_i)S_i) - \lambda\delta(x_i)S_i. \tag{A.1}$$

For fixed $\lambda \geq 0$, it is required to find the values of $x^*(\lambda)$ which maximize $F(x, \lambda)$ over $x \geq 0$. Since the Lagrangian is separable in the x_i variables, it suffices, for fixed λ, to find the values of $x_i^*(\lambda)$ which maximize $F_i(x_i, \lambda)$ over $x_i \geq 0$ for $i = 1, \ldots, K$.

For each i, $i = 1, \ldots, K$, it will be convenient to define the auxiliary function $G_i : R \times R^+ \to R$ as

$$G_i(x_i, \lambda) = (R_i - \lambda)x_i - C_i(x_i + S_i) - \lambda S_i. \tag{A.2}$$

The properties of G_i will be relevant to the maximization of F_i because of the following relation

$$F_i(x_i, \lambda) = \begin{cases} G_i(x_i, \lambda), & x_i > 0 \\ 0 & , x_i = 0. \end{cases} \tag{A.3}$$

*This was not possible in the model considered in [4].

We now consider the following possible ranges of λ values and derive the optimal plan $x_i^*(\lambda)$ for λ in each range.

CASE 1: $\lambda \geq R_i - C_i'(S_i)$. Recall that $R_i - C_i'(S_i) > 0$ by the nondegeneracy assumption of section 2.

CASE 2: $0 \leq \lambda < R_i - C_i'(S_i)$.

Case 1: $\lambda \geq R_i - C_i'(S_i)$

In this case it will be shown that $G_i(x_i, \lambda) < 0$ for every $x_i > 0$ and hence, from (3), it follows that $x_i^*(\lambda) = 0$. The proof follows from the concavity (in x_i) of $G_i(x_i, \lambda)$, which implies

$$G_i(x_i, \lambda) \leq G_i(0, \lambda) + \frac{d}{dx_i} G_i(0, \lambda) x_i = -C_i(S_i) - \lambda S_i + [R_i - \lambda - C_i'(S_i)] x_i,$$

which is negative for all $x_i > 0$ if $\lambda \geq R_i - C_i'(S_i)$. We thus have shown

$$\lambda \geq R_i - C_i'(S_i) \Rightarrow x_i^*(\lambda) = 0. \tag{A.4}$$

Case 2: $0 \leq \lambda < R_i - C_i'(S_i)$

In this case we first observe that $G_i(x_i, \lambda)$ assumes a unique global maximum at $\xi_i(\lambda) = c_i(R_i - \lambda) - S_i$, where c_i is the inverse function of C_i'. The reasoning is as follows. In section 1 it was assumed that $C_i(\xi)$ is a strictly convex function defined for all ξ, nonnegative and increasing for $\xi \geq 0$. Also, $C_i(0) = 0$, C_i is twice continuously differentiable, and $C_i'(x_i)$ is unbounded. It is thus apparent from (A.2) that, for fixed λ, $G_i(x_i, \lambda)$ is strictly concave in x_i. To find a unique global maximum of $G_i(x_i, \lambda)$ it is necessary and sufficient to find a solution to the equation

$$\frac{d}{dx_i} G_i(x_i, \lambda) = 0,$$

where λ is fixed and x_i is the variable.

It is thus required to solve

$$R_i - \lambda = C_i'(x_i + S_i). \tag{A.5}$$

Recall the Case 2 assumption that the range of λ is $0 \leq \lambda < R_i - C_i'(S_i)$. This implies $R_i - \lambda > C_i'(S_i)$. Since C_i' is unbounded and strictly increasing (from the strict convexity of C_i) it follows that there exists an x_i such that (A.5) holds. Using the property of the inverse function, c_i, we have

$$R_i - \lambda = C_i'(x_i + S_i) \Leftrightarrow c_i(R_i - \lambda) = c_i(C_i'(x_i + S_i)) \Leftrightarrow c_i(R_i - \lambda) = x_i + S_i.$$

That is, for each fixed λ, $0 \leq \lambda < R_i - C_i'(S_i)$, $G_i(x_i, \lambda)$ has a unique global maximum at the point

$$\xi_i(\lambda) = c_i(R_i - \lambda) - S_i.$$

Note that the Case 2 assumption that $0 \leq \lambda < R_i - C_i'(S_i)$ implies $R_i - \lambda > C_i'(S_i)$ which implies $c_i(R_i - \lambda) > S_i$ which implies $\xi_i(\lambda) > 0$. We can thus conclude from (A.3) that if $G_i(\xi_i(\lambda), \lambda) > 0$, $x_i^*(\lambda) = \xi_i(\lambda)$; if $G_i(\xi_i(\lambda), \lambda) < 0$, $x_i^*(\lambda) = 0$, and if $G_i(\xi_i(\lambda), \lambda) = 0$, $x_i^*(\lambda)$ can be taken as either $\xi_i(\lambda)$ or zero.

Now we show the existence of a $\hat{\lambda}_i$ in the open interval $(0, R_i - C_i'(S_i))$ with the properties

(i) $G_i(\xi_i(\lambda), \lambda) > 0$ if $0 < \lambda < \hat{\lambda}_i$,

(ii) $G_i(\xi_i(\hat{\lambda}_i), \hat{\lambda}_i) = 0$,

(iii) $G_i(\xi_i(\lambda), \lambda) < 0$ if $\hat{\lambda}_i < \lambda < R_i - C_i'(S_i)$.

Observe that the interval $(0, R_i - C_i'(S_i))$ is well defined since the nondegeneracy assumption implies that $R_i - C_i'(S_i) > 0$. In case (i) above, it then follows that

(A.6) $$0 < \lambda < \hat{\lambda}_i \Longleftrightarrow x_i^*(\lambda) = \xi_i(\lambda) = c_i(R_i - \lambda) - S_i.$$

In case (ii) above, it follows that

(A.7) $$\lambda = \hat{\lambda}_i \Rightarrow x_i^*(\lambda) = 0 \text{ or } x_i^*(\lambda) = \xi_i(\hat{\lambda}_i) = c_i(R_i - \hat{\lambda}_i) - S_i;$$

i.e., alternative optima exist. In case (iii) above, it follows from (A.3) that

(A.8) $$\hat{\lambda}_i < \lambda < R_i - C_i'(S_i) \Rightarrow x_i^*(\lambda) = 0.$$

For the case $\lambda = 0$, the nondegeneracy assumption implies $G_i(\xi_i(0), 0) > 0$, and hence $x_i^*(0) = \xi_i(0) = c_i(R_i) - S_i$. Combining (A.4), (A.6), (A.7), (A.8), with the case for $\lambda = 0$, the following rule has been established for optimizing $F_i(x_i, \lambda)$:

$$x_i^*(\lambda) = \begin{cases} c_i(R_i - \lambda) - S_i, & \text{if } 0 \leq \lambda < \hat{\lambda}_i \\ 0, \text{ or } c_i(R_i - \lambda) - S_i, & \text{if } \lambda = \hat{\lambda}_i \\ 0, & \text{if } \lambda > \hat{\lambda}_i. \end{cases}$$

This is Equation (2.1) in the main body of the exposition.

We now present the lemma which proves the existence of the number $\hat{\lambda}_i$ alluded to in the above remarks.

LEMMA: *Assuming* $R_i - C_i'(S_i) > 0$, *there is a* $\hat{\lambda}_i$ *in the open interval* $(0, R_i - C_i'(S_i))$ *such that the above relations* (i), (ii), *and* (iii) *are valid.*

PROOF: Let $\xi_i(\lambda) = c_i(R_i - \lambda) - S_i$ and for $\lambda \in [0, R_i - C_i'(S_i)]$ define the function $H_i(\lambda) = G_i(\xi_i(\lambda), \lambda)$. It is clear that $H_i(\lambda)$ is continuous, since it is the composite of continuous functions. Also $H_i(0) = G_i(\xi_i(0), 0) > 0$, by the nondegeneracy assumption, and $H_i(R_i - C_i'(S_i)) = -C_i(S_i) < 0$. Then by the continuity of H_i, there is a number $\hat{\lambda}_i \in (0, R_i - C_i'(S_i))$ such that

$$H_i(\hat{\lambda}_i) = 0 = G_i(\xi_i(\hat{\lambda}_i), \hat{\lambda}_i),$$

which proves (ii). In order to prove (i) and (iii), it is only necessary to demonstrate that $H_i(\lambda)$ is a decreasing function for $\lambda \in (0, R_i - C_i'(S_i))$. Let $0 < \lambda^1 < \lambda^2 < R_i - C_i'(S_i)$, and $\xi_i(\lambda^j) = c_i(R_i - \lambda^j) - S_i$, $j = 1, 2$. Then we have

$$\begin{aligned} H_i(\lambda^1) &= (R_i - \lambda^1)\xi_i(\lambda^1) - C_i(\xi_i(\lambda^1) + S_i) - \lambda^1 S_i \\ &> (R_i - \lambda^1)\xi_i(\lambda^2) - C_i(\xi_i(\lambda^2) + S_i) - \lambda^1 S_i \\ &> (R_i - \lambda^2)\xi_i(\lambda^2) - C_i(\xi_i(\lambda^2) + S_i) - \lambda^2 S_i = H_i(\lambda^2). \end{aligned}$$

The first inequality follows from the fact that $\xi_i(\lambda_1)$ is the unique value of x_i that maximizes $G_i(x_i, \lambda^1)$. The second inequality is due to the fact that $\lambda^1 < \lambda^2$ and $\xi_i(\lambda^j) > 0$, $j = 1, 2$. This establishes the monotonicity of H_i and completes the proof of the lemma.

REFERENCES

[1] Bellmore, M., H. J. Greenberg, and J. J. Jarvis, "Generalized Penalty-Function Concepts in Mathematical Optimization." Operations Research, Vol. 18, No. 1 (1970).

[2] Brooks, R. and A. Geoffrion, "Finding Everett's Lagrange Multipliers by Linear Programming." Operations Research, Vol. 14, No. 6 (1966).

[3] Evans, J. P., and F. J. Gould, "Stability in Nonlinear Programming." Operations Research, Vol. 18, No. 1 (1970).

[4] Evans, J. P., and F. J. Gould, "An Exhaustive Sweep GLM Algorithm for Production Scheduling," to appear in Management Science.

[5] Everett, H., "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources." Operations Research, Vol. 11, No. 3 (1963).

[6] Gould, F. J., "Extensions of Lagrange Multipliers in Nonlinear Programming," SIAM, Vol. 17, No. 6 (1969).

SEQUENCING MANY JOBS ON A MULTI-PURPOSE FACILITY

John A. Buzacott

and

Sujit K. Dutta

Department of Industrial Engineering
University of Toronto
Toronto, Canada

ABSTRACT

Suppose a given set of jobs has to be processed on a multi-purpose facility which has various settings or states. There is a choice of states in which to process a job and the cost of processing depends on the state. In addition, there is also a sequence-dependent changeover cost between states. The problem is then to schedule the jobs, and pick an optimum setting for each job, so as to minimize the overall operating costs.

A dynamic programming model is developed for obtaining an optimal solution to the problem. The model is then extended using the method of successive approximations with a view to handling large-dimensioned problems. This extension yields good (but not necessarily optimal) solutions at a significant computational saving over the direct dynamic programming approach.

INTRODUCTION

This paper deals with the problem of scheduling N jobs on a single 'multi-purpose' facility which has various adjustable settings that can process a variety of jobs. Let us define a 'state' as a unique arrangement of those settings. There is a choice of states in which to process a job and the cost of processing depends on the state. In addition, there is also a sequence-dependent changeover cost between states. The problem is then to schedule the jobs, and pick an optimum setting for each job, so as to minimize the overall operating costs.

Burstall [3] has reported that this problem was encountered in the manufacture of steel tubes. Another area where one is frequently faced with this type of problem is in a machine shop. Typically this consists of a number of machines some of which are likely to be multi-purpose (otherwise called general-purpose) machines. In many real life machine shops these multi-purpose machines work quite independently of one another, and in those cases it is possible to break down the whole problem of scheduling all the machines into independent sub-problems, each pertaining to a different multi-purpose machine. These subproblems then resemble our scheduling model.

CHARACTERISTICS OF THE PROBLEM

The 'sequencing' problem may be characterized by the following assumptions:

1. There are N given jobs that are to be processed.

2. There is only one multi-purpose machine (facility) available and it has M different 'states'. At a given time the machine cannot be in more than one state. Moreover, a state can handle only one job at a time.

3. A job is completed once it has been processed by a single state.

4. Each job can be processed by the machine in at least one of the states and, additionally, it may be possible to process more than one of the jobs with a single state.

5. All jobs are considered equal in importance. Thus there are no due dates, priorities, or rush orders.

6. An operation once started must be completed without interruption.

7. There are only two types of costs:

a) c_{ij}= Processing cost of job i at state j. (If it is infeasible to process job i at state j, the symbol $c_{ij}=\infty$ is used.)

b) h_{pq}= Changeover cost from state p to state q. (h_{pq} and h_{qp} might have different values.) All costs are assumed to be known without error.

The objective is to find a solution as a sequence, Q, of states and an assignment, A, of jobs to states which satisfies above assumptions and minimizes:

$$Z=\sum c_{ij}+\sum h_{pq},$$

where the first summation is over all pairs (i, j) such that i is assigned to j in A, and the second summation is over all pairs (p, q) such that p immediately preceeds q in Q, $i\epsilon J$; $j, p, q\epsilon S$, where J and S are the sets of all given jobs and states, respectively.

This optimization problem and variations of it actually arise in many industrial situations. The model also fits situations that are entirely different. A typical example appears below.

THE DECORATOR'S PROBLEM

A decorator wants to purchase N different articles for meeting his present contracts. There are M different stores situated in M different locations, from which the articles may be bought. Let h_{pq} be the cost of travelling from store p to store q (note that h_{pq} is not necessarily equal to h_{qp}) and c_{ij} be the price of article i at store j. If an article i is not available at store j, the symbol $c_{ij}=\infty$ is used. The cost matrices C_{ij} and H_{pq} are assumed to be known beforehand. The problem is to find which stores are to be visited, in what sequence, and which article or articles should be bought in each so as to minimize the overall cost resulting from purchase and travel.

If the state for each job is fixed, (i.e., $c_{ij}=\infty$ for all $j\epsilon S$ except $j=j(i)$, where S is the set of all given states and $j(i)$ is the only feasible state where job i can be processed), then the total processing cost is a constant. The problem of minimizing only changeover cost among the chosen set of states is a travelling salesman problem (see [2]). Thus our problem can be thought of as a generalization of the travelling salesman problem in which the order of a state as well as the job (jobs) to be done in that state must be specified.

Another machine sequencing problem related to the travelling salesman problem is that considered by Gilmore and Gomory [4]. There are N jobs to be sequenced on a machine having a state described by a single real variable x. Each job has two associated numbers A_i and B_i as its starting and ending states respectively. If job j follows job i, the state of the machine must then be changed from B_i to A_j and the cost of this change is c_{ij}. Note that in Gilmore and Gomory's problem there is no choice of states for a given job.

SEARCH FOR OPTIMAL SOLUTIONS

Burstall [3] has given a heuristic solution for this problem which works reasonably well for the particular case where the differences in the production times (costs) for a given job in various states are small compared with the changeover times (costs) between states. Burstall proposed a branch-and-bound technique as the search procedure for which Lomnicki [5] has suggested a simpler method

using Boolean algebra. Note that none of these methods can guarantee optimal solutions. Also when the cost matrices are not of the desired type their methods yield solutions that are far from the true optimal solutions. However, no trace of any other solution schemes for this problem has been found.

With a view to finding an exact solution to the problem we developed a zero-one integer programming formulation. But computationally, this approach did not seem to be very useful. This was mainly because a large number of subtour constraints were necessary even for a reasonably small sized problem. Moreover the computer time required to identify an optimal solution was, in general, quite high. As mentioned above, this problem resembles a generalized version of the travelling salesman problem, for which it is known that the integer programming solutions are not particularly suitable [2]. In their survey of the travelling salesman problem, Bellmore and Nemhauser [2] have reported that for problems of less than or equal to 13 cities they would use dynamic programming. So it is logical to think that an appropriate dynamic programming scheme might be suitable for solving 'sequencing' problems of at least small dimensions. The expected limitations of a dynamic programming approach is usually due to the curse of dimensionality. It will be shown in a later section, however, how this difficulty can be surmounted by using the technique of successive approximations.

A DYNAMIC PROGRAMMING MODEL

A step wise dynamic programming formulation that obtains an optimal solution to the problem is discussed below.

1. *Stage Variable:* K — Kth choice of job has to be made; $K-1$ jobs have already been processed. $K=1, 2, \ldots, N$.

2. *State Variables:*

   i) i_K — the machine state used in stage $K-1$. $i_K = 1, 2, \ldots, M$. (If a starting state is not given imagine $i_1 = D$ — a ficticious state, s.t. $h_{D,i} = 0 \forall i$).

   ii) y_K — the subset of jobs already processed, i.e., $y_K = \{x_1, x_2, \ldots, x_{K-1}\}$.

3. *Decision Variables:*

   i) x_K — the job to be processed at stage K. $x_K = 1, 2, \ldots, N$.

   ii) i'_K — the machine state to be used at stage K. $i'_K = 1, 2, \ldots, M$.

4. *Transition Relations:*

   i) $y_{K+1} = y_K U x_K$.

   ii) $i_{K+1} = i'_K$.

5. *Constraint:* $x_K \notin y_K = \{x_1, \ldots, x_{K-1}\}$.

6. *Economic Function:* Minimize $Z = \sum_{K=1}^{N} \{c_{x_K, i'_K} + h_{i_K, i'_K}\}$

where c_{ij} = cost of processing job i at state j.

h_{pq} = changeover cost from state p to state q.

7. *Recurrence Relation:*

$$f_K(i_K, y_K) = \underset{x_K, i'_K}{\text{Min}} \{C_{x_K, i'_K} + h_{i_K, i'_K} + f_{K+1}(i'_K, y_K U x_K)\}$$

$f_K(i_K, y_K)$ may be defined as the minimum scheduling cost starting with the machine in machine state i_K, when the jobs that have already been processed are known (through y_K).

IMPLEMENTING THE MODEL IN A COMPUTER

The state variable y_K may be represented by an N-dimensional vector (V) each element of which corresponds to a unique job. Initially, before any job has been processed, all the elements of V are made equal to zero. As soon as a job is processed the corresponding element in V is given a value of one. Thus y_K will be represented by a vector whose $K-1$ elements have a value of one and the remaining $N-K+1$ elements are zero. Note that there are $\binom{N}{K-1}$ of such vectors, only one of which represents y_K.

In order to reduce the cost of storing all these vectors, we represent each vector as a binary number and store the corresponding decimal number in lieu of storing the whole vector. There will be as many as 2^n-1 different vectors corresponding to 2^n-1 possible values of the state variables $y_i (i=1, 2, \ldots, n)$ when the number of jobs in a problem is n. These vectors may be treated as binary numbers and represented by the decimal numbers $0, 1, 2, \ldots, 2^n-2$. Unfortunately, the decimal numbers that correspond to the possible values of the state variable (y_K) at stage K are, in general, widely scattered. This creates a problem as to how to collect the numbers corresponding to the possible state variables at a particular stage. This may, however, be done by arranging the decimal numbers in such a way that the number of 1's in their binary representations are in non-increasing order. The first $\binom{N}{N-1}$ numbers from highest down would represent the domain of y_N, the following $\binom{N}{N-2}$ numbers would represent the domain of y_{N-1} and so on.

Example 1:

If $n=3$ the numbers would be arranged in the following way.

Stage K	State variable y_K	Binary Number			n_i^*	Decimal Equivalent
3	y_3	1	1	0	2	6
		1	0	1	2	5
		0	1	1	2	3
2	y_2	1	0	0	1	4
		0	1	0	1	2
		0	0	1	1	1
1	y_1	0	0	0	0	0

*n_i = no. of 1's in the binary representation.

This concept may be applied to the (first) transition relation as well. At stage K, if the ith job is processed we put a 1 in the column i of the vector that represents y_K to obtain a new vector for y_{K+1}. The equivalence of this operation in our representation is simply to add the number 2^{i-1} to the decimal representation of y_K to get a new decimal number which will now represent y_{K+1}.

Example 2:

$$\text{Let } y_2 = 1 \text{ and } i = 3$$

Then $y_3 = y_2 + 2^{i-1} = 1 + 2^{3-1} = 5$ and the binary representation (101) of the decimal number 5 does confirm that the jobs 1 and 3 are over.

LIMITATIONS

The algorithm was tested on an IBM Model 360/65 and was found to work well for small and medium sized problems. Burstall's [3] illustrative example of 8 jobs and 19 states was solved in less than 19 seconds of CPU time with a core requirement of less than 100K bytes. Unfortunately due to computational limitations the use of the dynamic program becomes restricted to problems having 12 jobs or less and any practical number of states. Nevertheless, with the use of disc storage it will be possible to handle problems of up to 15 jobs with a reasonable number of states. The dimensions of many practical problems (as quoted by Burstall from factory records) may be expected to be reasonably small so that an exact optimal solution may be obtained by the above formulations. In view of this, the restriction on the number of jobs does not appear to be discouragingly stringent; however, the following alternative solution scheme has been developed to tackle problems with a greater number of jobs.

EXTENSIONS

In order to overcome the dimensionality difficulty we propose to use the method of successive approximations [1, p. 78]. Starting with a known feasible solution, one way to employ this approach is to consider initially a subproblem of sequencing only p-jobs (p is a suitably chosen number $\leq N$) to all the available states, and solve the subproblem while the remaining jobs are kept fixed in the schedule. Then another subproblem with a different set of p-jobs is chosen to be solved and the process is continued until it converges. The choice of the sets of p-jobs may be done in various ways among which two popular ones are neighbour combinations and stochastic (or random) combinations. In this problem, however, the stochastic choice of the subsets of jobs would not be particularly suitable because the changeover costs among the states may be sequence-dependent.

Details of the Method

The detail workings of the method for an N-job M-state problem may now be summarized in the following steps.

1. Generate an initial feasible solution. (Burstall's and Lomnicki's methods may be used for this purpose.)

2. Choose a suitable value for p. (An alternative is to try with different values of p.)

3. From the initial schedule consider the first p jobs and solve the subproblem of scheduling these p jobs to all the M states, i.e., find a state for each of these p jobs and an optimal sequence of these p jobs. The starting and ending states for this subproblem should be found from the initial solution obtained in (1). This represents one 'step.' (The dynamic programming formulation may be used for solving these subproblems.)

4. To continue keep the same sequence and assignment (of states) for $(p+2)$, $(p+3)$, . . . , Nth jobs as before, use the assignment (of a state) to the first job as determined by the previous step, and then solve the new subproblem for the 2nd, 3rd, . . . , $(p+1)$th jobs.

5. Continue in this way until all the jobs have been considered in at least one of the subproblems. This represents one 'iteration.' Then calculate the total cost of the schedule thus obtained. If there is any improvement continue with another iteration, otherwise halt because any further iterations cannot produce a better solution.

A Note on the Selection of the Value of p

It has been experienced that increasing the value of p increases the computational time and core

required, with an improvement in the chance of getting accurate results. Therefore the choice of p should be made by trading off these factors, i.e., for a reasonable and quick solution a low value of p should be specified whereas higher values of p should be used for getting precise solutions. For quickly obtaining a reasonably good solution a recommendable value of p seems to be 4 or 5. Nevertheless, it depends to a great extent on the dimension of the problem to be solved and also on the accuracy sought.

Choosing the Starting Solution

The efficiency of the method of successive approximations rests very largely on the selection of the starting solution. As it is not possible, in general, to predict the best starting solution, we recommend to repeat the proposed search-procedure with different starting solutions. The number of starting solutions to be used for a given problem again depends on the accuracy sought.

Note that any feasible solution may be used as starting solution. In successive approximations technique the value of the objective function of the starting solution has no influence on that of the final solution. We have recommended Burstall's and Lomnicki's methods to generate the starting solutions only because they are simple and independent of the actual cost values.

Computational Experience

A computer program was written in FORTRAN IV (Level G) and tested with a number of small and medium sized problems for which the true optimal solutions were evaluated using the direct d.p. approach. Although this method could not generate the exact optimal solution in every case, the solutions, in general, were within 10 percent of the true optimal value. Moreover it has been observed in many cases that if different starting solutions are used then the method yields exact optimal solution even with a very low value (viz. 3) of the step size, p. Also the computational time in this method is much less compared to that for the direct d.p. method. For example, a 10-job, 10-state sample problem was solved by this method in less than 25 seconds of CPU time [and the cost of the solution was 49], whereas exactly the same solution was obtained by the direct d.p. approach in about 144 seconds (and the heuristic solution due to Lomnicki and Burstall yielded a cost of 272).

To demonstrate the sensitivity of the final solution with respect to the starting solution, the above sample problem was solved with different starting solutions. The results are shown in Table 1. Every run was made on an IBM Model 360/65 and in every case a step size of 3 was used.

TABLE 1. *Sensitivity of the Final Solution with respect to the Starting Solution*

Solution No.*	sol	fsl	it	Execution time (CPU) in secs
1	[illegible]	60	3	[illegible]
2	272	66	3	3
3	[illegible]	58	3	[illegible]
4	[illegible]	58	[illegible]	[illegible]
5	[illegible]	50	2	2
6	[illegible]	[illegible]	3	3
7	[illegible]	49	2	2
8	[illegible]	[illegible]	2	2
9	[illegible]	[illegible]	2	2
10	[illegible]	49	2	2

* All the ten starting solutions were generated using Burstall's and Lomnicki's approaches.

Key:

iof – objective function of the starting solution.

fof – objective function of the final solution.

it – number of iterations performed.

Table 2 supports the hypothesis that increasing the value of p increases the chance of getting better results at the expense of computational effort. Solution No. 2 in Table 1, (which happened to be Burstall's and Lomnicki's solution to this problem) was used as the starting solution in every case. As before, the runs were made on an IBM Model 360/65.

TABLE 2. *Sensitivity of the Final Solution with respect to the Step Size*

Run No.	p	*of*	*it*	Execution time [illegible] in secs
1	3	340	3	2
2	4	335	3	5
3	5	333	2	17
4	6	332	[illegible]	25
5	7	332	[illegible]	34

Key:

p – step size.

of – objective function of the final solution.

it – number of iterations performed.

CONCLUDING COMMENTS

The problem of sequencing many jobs on a multi-purpose facility has been solved by a dynamic programming algorithm. Then the technique of successive approximations has been successfully applied to extend the d.p. model. This extended method does not impose any restriction on the size of the problem and so provides answers to those large dimensional problems which cannot be directly handled by the dynamic programming scheme. Besides, for a given problem this method needs considerably less computational effort compared to that for the other method. Thus even for a problem which can be solved by the other method one may prefer this approach. This would be the case when one is interested in getting a reasonably good solution with little computer effort, as opposed to obtaining a guaranteed optimal solution at the expense of a comparatively large amount of computational time. Finally, it may be hoped that this study will encourage other applications of the method of successive approximations.

ACKNOWLEDGEMENTS

The authors wish to thank Professor P. E. [illegible] for his helpful suggestions. The financial assistance provided through the National Research Council of Canada is also gratefully acknowledged.

REFERENCES

[1] Bellman, R. E., and S. E. Dreyfus, *Applied Dynamic Programming* (Princeton University Press, N.J., 1962).

[2] Bellmore, M., and G. L. Nemhauser, "The Traveling Salesman Problem: A Survey," Operations Research *16*, 538–558 (May 1968).

[3] Burstall, R. M., "A Heuristic Method for a Job-Scheduling Problem," Operational Research Quarterly *17*, 291–304, No. 3 (Sept. 1966).

[4] Gilmore, P. C., and R. E. Gomory, "Sequencing a One-State Variable Machine: A Solvable Case of the Travelling Salesman Problem," Operations Research, *12*, 655–679, No. 5 (Sept. 1964).

[5] Lomnicki, Z. A., "Job Scheduling," Letters to the Editor, Operational Research Quarterly, *17*, 314–317 (Sept. 1966).

LOCATION OF FACILITIES WITH RECTANGULAR DISTANCES AMONG POINT AND AREA DESTINATIONS*

G. O. Wesolowsky

McMaster University
Hamilton, Ontario

and

R. F. Love

University of Wisconsin
Madison, Wisconsin

ABSTRACT

This article is concerned with the optimal location of any number (n) of facilities in relation to any number (m) of destinations on the Euclidean plane. The criterion to be satisfied is the minimization of total weighted distances where the distances are rectangular. The destinations may be either single points, lines or rectangular areas. A gradient reduction solution procedure is described which has the property that the direction of descent is determined by the geometrical properties of the problem.

1. INTRODUCTION

Many contributions to the literature on location models have dealt with variations and extensions of the Weber problem (for example: [8] and [12]). The object in such problems is to locate one or more points (facilities) on a plane that contains fixed points (destinations). Optimum location is achieved when the sum of weighted distances between points in the system is minimized. Such models have found varied applications including some in plant location [2] and in communication networks [10].

In many of these models the distances between points are designated to be Euclidean straight line distances; however, in the context of location in a grid of city streets or in a network of aisles in a factory or warehouse, rectangular distances often constitute a better approximation to actual distances and have been used in location models (for example: [6] and [11]). The model in this paper is a multi-facility model. Shipments between the facilities to be located are possible. This is a variation of the trans-shipment problem in linear programming. Previous models including inter-facility flows are given in [1], [3–5], [9], [10], and [14].

Considering each destination as a separate point is often impractical when large populations like those in cities are involved. A useful approximation is to consider destinations to be uniformly distributed over some area or areas. This approximation has been used in models involving postal districts [1] and facility design [7]. The inclusion of this approximation in multi-facility models allows the solution of very large and spatially complex location systems.

*This research was supported by a grant from the Control Data Corporation. Use of the University of Wisconsin Computing Center was made possible through support, in part, from the National Science Foundation, other United States Government agencies and the Wisconsin Alumni Research Foundation through the University of Wisconsin Research Committee.

This paper presents a method for the optimal location of one or more facilities among any number of destinations which can be points or areas. The areas are restricted to be rectangles with sides parallel to the axes; overlapping is allowed. All distances in the system are rectangular.

2. THE LOCATION OF ONE FACILITY

Before discussing the more general case of the location of multiple facilities a method for locating a single facility is presented. Figure 1 shows an example with five destination points and three destination areas. It is necessary to locate a point (facility) such that the weighted sum of distances in the system is minimized.

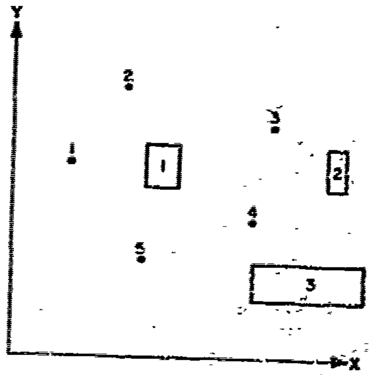


FIGURE 1.

Let the coordinates of the facility be (x, y). There are m point destinations located at (a_i, b_i), $i=1, \ldots, m$. Each of the p area destinations A_k, $k=1, \ldots, p$, can be completely identified by its four corner points (c_k, e_k), (c_k, f_k), (d_k, f_k), (d_k, e_k), where $d_k > c_k$, $f_k > e_k$.

The location problem now consists of minimizing

(1) $$W_{1mp}(x, y)=\sum_{i=1}^{m} w_i(|x-a_i|+|y-b_i|)+\sum_{k=1}^{p} u_k \int\int_{A_k} (|x-z_1|+|y-z_2|)\,dz_1dz_2.$$

The weights w_i transform distances into costs. The constants u_k can combine factors of cost per unit distance and population density.

It should be noted that $w_{1m0}(x, y)$ (the case where $p=0$) has been minimized by Francis [4]. Francis made use of the fact that $w_{1m0}(x, y)$ is separable in x and y to achieve a minimum by optimizing with respect to one variable at a time. That is, the optimum value of x can be obtained by minimizing $\sum_{i=1}^{m} w_i|x-a_i|$ and the optimum value of y can be obtained by minimizing $\sum_{i=1}^{m} w_i|y-b_i|$.

The function $w_{1mp}(x, y)$ is also separable in x and y and the optimum value of x can be obtained by minimizing

(2) $$WX_{1mp}(x)=\sum_{i=1}^{m} w_i|x-a_i|+\sum_{k=1}^{p} u_k(f_k-e_k)\int_{c_k}^{d_k} |x-z_1|\,dz_1.$$

The minimization of $WX_{1mp}(x)$ can be thought of as the location of a facility among destinations placed on an axis; the destinations are points and lines. The destinations could overlap.

It is convenient to pose an equivalent problem where none of the destinations overlap. This can readily be done.

For example, if there are two line destinations $[c_r, d_r]$ and $[c_s, d_s]$, such that $c_r < c_s$, $c_s < d_r$ and $d_s > d_r$, then

$$U_r \int_{c_r}^{d_r} |x - z_1|\, dz_1 + U_s \int_{c_s}^{d_s} |x - z_1|\, dz_1 = U_r \int_{c_r}^{c_s} |x - z_1|\, dz_1$$

$$+ (U_r + U_s) \int_{c_s}^{d_r} |x - z_1|\, dz_1 + U_s \int_{d_r}^{d_s} |x - z_1|\, dz_1,$$

where $U_k = u_k(f_k - e_k)$, and three non-overlapping lines $[c_r, c_s]$, $[c_s, d_r]$, and $[d_r, d_s]$ are obtained. Similarly, if a point destination is within a line destination, the overlap can be removed by dividing the line into two at that point.

After the overlaps have been removed there is some number p' of line destinations, where $p' \geqslant p$. Combine any point destinations with the same location on the x axis to obtain $m'(m \leqslant m)$ points.

Arrange the destinations along the axis from left to right and label them $[r_k, s_k]$, $k = 1, \ldots, p' + m'$, where $r_{k+1} \geqslant s_k$ and $r_k \leqslant s_k$. If the kth destination is a line then let $c_k' = r_k$, $d_k' = s_k$ and the constant be U_k'; if the kth destination is a point then $a_k' = r_k = s_k$ and the corresponding weight is w_k'.

After the removal of overlaps, $WX_{1mp}(x)$ has $p' + m'$ terms. A term, such as $w_k'|x - a_k|$ (and its derivative), is plotted in Figure 2. If the destination is a line $[c_k', d_k']$, the corresponding term and its derivative are as in Figure 3. Since the terms are obviously convex, then $WX_{1mp}(x)$ and $W_{1mp}(x, y)$ are convex.

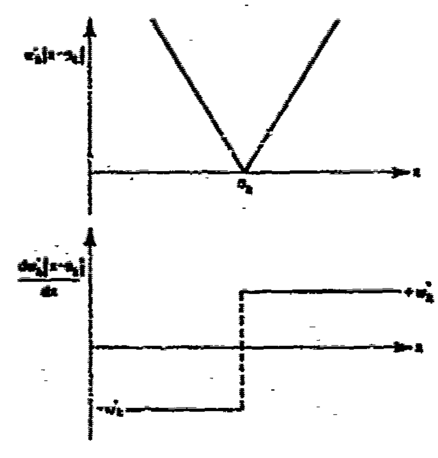


FIGURE 2.

From Figures 2 and 3 it is evident that when $x < r_k$ the slope of the term corresponding to $[r_k, s_k]$ is negative and a constant and when $x > s_k$ the slope is the positive value of that constant. It is possible to evaluate $dWX_{1mp}(x)/dx$ at the values s_k, $k = 1, \ldots, m' + p'$.

Let the absolute value of the constant slope corresponding to $[r_k, s_k]$ be t_k, $k = 1, \ldots, m' + p'$. For a point destination t_k is the weight w_k' and for a line destination it is the constant $U_k'(d_k' - c_k')$.

When $x < r_1$

$$\frac{dWX_{1mp}(x)}{dx} = -\sum_{i=1}^{p'+m'} t_i = -M.$$

Consequently,

$$\left.\frac{dWX_{1mp}(x)}{dx}\right|_{s_k} = -M + 2\sum_{j=1}^{k} t_j. \tag{3}$$

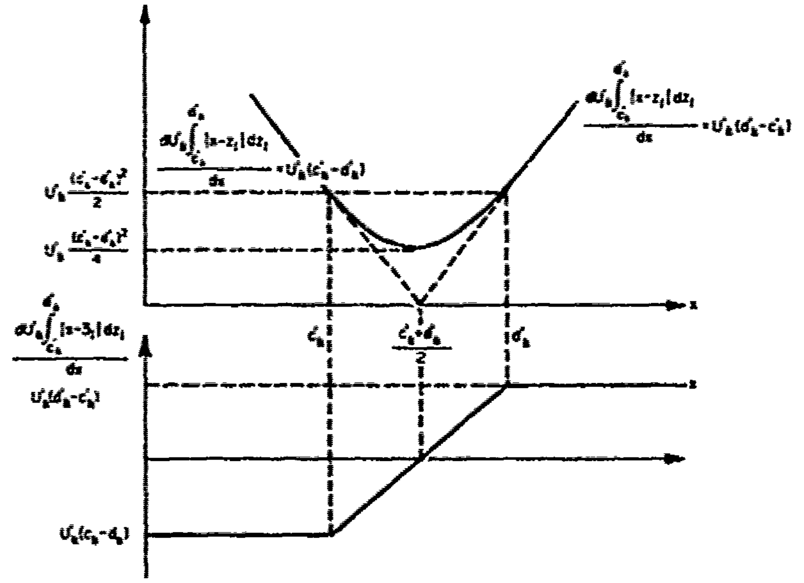


FIGURE 3.

Since $WX_{1mp}(x)$ is convex, (3) enables us to find the region in which $WX_{1mp}(x)$ is a minimum. Expression (3) is evaluated for successively larger integers, k, until it becomes either zero or positive. If it becomes zero for some value of k, say k', then $s_{k'} \leq x^* \leq s_{k'+1}$, where x^* is the optimum value of x. If expression (3) becomes positive for the first time when $k = k'$, then $r_{k'} \leq x^* \leq s_{k'}$. In this latter case, if region k is a line, the exact position of x^* can be found by using the derivative plotted in Figure 3.

They coordinate, y^*, of the optimum location for the facility can be found in an entirely similar manner. The following example illustrates the method discussed above.

Example:

$$WX_{112}(x) = 2|x-2| + 2\int_1^4 |x-z_1|dz_1 + 1\int_3^6 |x-z_1|dz_1.$$

Minimizing $WX_{112}(x)$ with respect to x corresponds to finding the optimum x coordinate for a facility which has one point destination ($m=1$) and two area destinations ($p=2$).

First remove the overlaps.

$$WX_{112}(x) = 2\int_1^2 |x-z_1|dz_1 + 2\cdot|x-2| + 2\int_2^3 |x-z_1|dz_1 + 3\int_3^4 |x-z_1|dz_1 + 1\cdot\int_4^6 |x-z_1|dz_1.$$

Then $m' = 1$, $p' = 4$, and

$$r_1 = 1,\ s_1 = 2,\ U'_1 = 2,\ t_1 = 2,$$

$$r_2 = 2,\ s_2 = 2,\ w'_2 = 2,\ t_2 = 2,$$

$$r_3 = 2,\ s_3 = 3,\ U'_2 = 2,\ t_3 = 2,$$

$$r_4 = 3,\ s_4 = 4,\ U'_4 = 3,\ t_4 = 3,$$

$$r_5 = 4,\ s_5 = 6,\ U'_2 = 1,\ t_5 = 2,$$

and

$$M = (2+2+2+3+2) = 11.$$

From (3)

$$\left.\frac{dWX_{112}(x)}{dx}\right|_{r_1^*}=-11+4=-7.$$

$$\left.\frac{dWX_{112}(x)}{dx}\right|_{r_2^*}=-11+4+4=-3.$$

$$\left.\frac{dWX_{112}(x)}{dx}\right|_{s_2^*}=-11+4+4+4=1.$$

$$\therefore r_3' \leq x^* \leq s_2.$$

The slope of the third term must be 1 at x^*. From Figure 3

$$1=U_3'(2x^*-(c_3'+d_3')).\ \text{and}\ x^*=\frac{11}{4}.$$

3. THE TWO FACILITY PROBLEM

Consider now the problem of locating two facilities (x_1, y_1) and (x_2, y_2) among p area destinations and m point destinations. It is necessary to minimize

$$\text{(4)}\qquad W_{2mp}(x_1, y_1, x_2, y_2)=\sum_{j=1}^{2}\sum_{i=1}^{m} w_{ji}(|x_j-a_i|+|y_j-b_i|)$$
$$+\sum_{j=1}^{2}\sum_{k=1}^{p} u_{jk}\int_{A_k}\int(|x_j-z_1|+|y_j-z_2|)\,dz_1dz_2+v_{12}(|x_1-x_2|+|y_1-y_2|).$$

The weight, w_{ji}, applies to the distance between facility, j, and point destination, i, while u_{jk} applies to the distance between facility, j, and area, k. The interfacility distance is weighted by v_{12}.

As before, the finding of the x coordinates for the optimal location of the facilities is independent of finding the y coordinates.

Consider the minimization of

$$\text{(5)}\qquad WX_{2mp}(x_1, x_2)=\sum_{j=1}^{2}\sum_{i=1}^{m} w_{ji}|x_j-a_i|+\sum_{j=1}^{2}\sum_{k=1}^{p} u_{jk}\cdot(f_k-e_k)\int_{c_k}^{d_k}|x_j-z_1|\,dz_1+v_{12}|x_1-x_2|.$$

In the case $p=0$ (which, incidentally, was solved graphically by Francis [4]), (5) is a polyhedral surface in (WX_{2mp}, x_1, x_2) space. This is true because terms such as $w_{ji}|x_j-a_i|$ and $v_{12}|x_1-x_2|$ are made up of half-planes. When $p\neq 0$ some of the "edges" of the polyhedral surface become parabolically "rounded" as can be deduced from Figure 3.

Let P be the set of projections of the edges on the x_1x_2 plane. Then P is given by

$$\text{(6)}\qquad P=\bigcup_{i=1}^{3} P_i \text{ where}$$

$$P_1=\{x_1, x_2 \mid x_1=x_2\},$$
$$P_2=\{x_j \mid x_j=a_i\},\qquad j=1, 2;\qquad i=1, \ldots, m', \text{ and}$$
$$P_3=\{x_j \mid c_k'\leq x_j\leq d_k'\},\qquad j=1, 2;\qquad k=1, \ldots, p'.$$

Assume that any projections corresponding to zero weights are omitted (i.e., if $w_{ji}=0$, omit projection $x_j=a_i$).

Those points on the surface of $WX_{2mp}(x_1, x_2)$ which are not in P are located on plane sections since the direction numbers of tangent planes can only change in P. A minimum cannot occur on a olane without occurring on at least one of the edges.

The search for the minimum of $WX_{2mp}(x_1, x_2)$ need take place only among the projections. P. A relaxation or univariate method [13] is used to find this minimum. The procedure is as follows (see Figure 4).

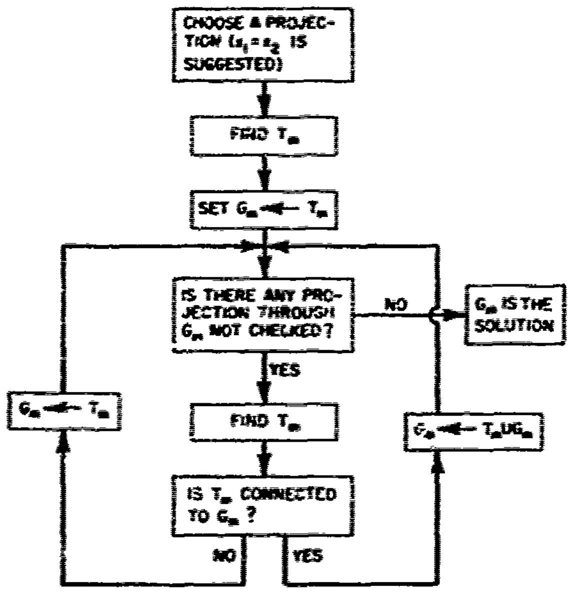


FIGURE 4.

Let T_m be the minimum of the cross-section of $WX_{2mp}(x_1, x_2)$ formed by a plane $x_j = h$ (h is any constant) or $x_1 = x_2$. T_m can be found by the method of the previous section. Let G_m be a set of points that all yield the same value of $WX_{2mp}(x_1, x_2)$: at the end of the algorithm the set G_m will contain the points that define the global minimum of $WX_{2mp}(x_1, x_2)$.

T_m is first found for an arbitrary projection. Let $G_m \leftarrow T_m$. All lines of the form $x_j = h$ or $x_1 = x_2$ that pass through T_m are investigated until points not in G_m are produced. Again $G_m \leftarrow T_m$ and the process is continued. When no new points can be generated a global minimum has been obtained. Local minima are precluded by the fact that $WX_{2mp}(x_1, x_2)$ is convex. An example illustrates the algorithm.

EXAMPLE:

$$WX_{2mp}(x_1, x_2) = 1 \cdot |x_1 - x_2| + 2 \cdot \int_1^2 |x_1 - z_1| \, dz_1 + 1 \cdot \int_5^6 |x_1 - z_1| + 1 \cdot \int_1^2 |x_2 - z_2| \, dz_2 + 3 \cdot \int_5^6 |x_2 - z_2| \, dz_2.$$

P is given by Figure 5.

Let $x_1 = x_2$ and find T_m

$$T_m = \left\{ x_1, x_2 \mid x_1 = x_2 = 5\frac{1}{8} \right\}.$$

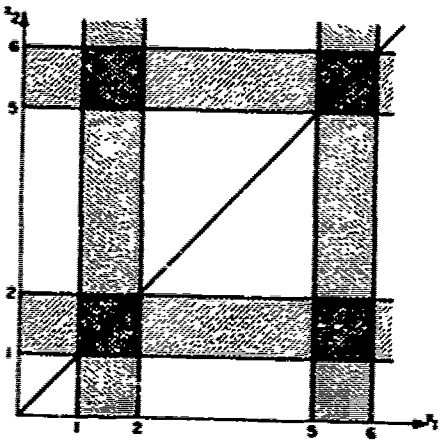


FIGURE 5.

Let $G_\omega \leftarrow T_\omega$.

Now check the line $x_2 = 5\text{-}1/8$ which passes through G_ω and is in P. The result is

$$T_\omega = \left\{x_1, x_2 \mid 2 \leq x_1 \leq 5,\ x_2 = 5\frac{1}{8}\right\}.$$

Let $G_\omega \leftarrow T_\omega$.

Next consider the line $x_1 = 2$.

$$T_\omega = \left\{x_1, x_2 \mid x_1 = 2,\ x_2 = 5\frac{1}{6}\right\}.$$

$T_\omega \leftarrow T_\omega$.

Consider the line $x_2 = 5\text{-}1/6$ passing through G_ω.

$$T_\omega = \left\{x_1, x_2 \mid 2 \leq x_1 \leq 5,\ x_2 = 5\frac{1}{6}\right\}.$$

$G_\omega \leftarrow T_\omega$.

Further steps yield no change in G_ω and hence x_1^* lies in the interval [2, 5], and $x_2^* = 5\text{-}1/6$.

Computational experience suggests that two facilities can be located with the investigation of less than 20 t_ω's even for problems with large p's and m's.

4. THE N FACILITY PROBLEM

The general problem of locating n facilities $(x_1, y_1), \ldots, (x_n, y_n)$ among m destination points and p destination areas can be written as the minimization of

$$(7)\qquad W_{\exp}(x_1, y_1, \ldots, x_n, y_n) = \sum_{j=1}^{n}\sum_{i=1}^{m} w_{ji}(|x_j - a_i| + |y_j - b_i|)$$
$$+ \sum_{j=1}^{n}\sum_{k=1}^{p} u_{jk} \iint_{A_k} (|x_j - z_1| + |y_j - z_2|)\,dz_1 dz_2 + \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} v_{jk}(|x_j - x_k| + |y_j - y_k|).$$

where the v_{jk} are interfacility weights. The finding of the optimal x coordinates can again be carried out separately.

The univariate method of optimization applies, but is obviously very tedious when n is large. It is, of course, possible to program a computer to do the calculations.

5. CONCLUSIONS

Since points and overlapping rectangular areas can be used to approximate quite complex spatial distributions of populations the methods discussed above would seem to have applicability in urban location. Warehouse and plant layouts are often comprised of aisles laid out in a grid. If the grid is fine enough rectangular distances would apply and the models discussed above could be used to solve some material flow problems (see [4] and [6]).

REFERENCES

[1] Bender, B. K. and A. J. Goldman, "Optimization of Distribution Networks," *NBS Rept* 6930 (1962).

[2] Burstall, R. M., R. A. Leaver, and J. E. Sussands, "Evaluation of Transport Cost for Alternative Factory Sites," Operational Resaerch Quarterly *13*, 345–354 (1962).

[3] Cabot, A. V., R. L. Francis, and M. A. Stary, "A Network Flow Solution to a Rectilinear Distance Facility Location Problem," AIIE Transactions *2*, 132–141 (1970).

[4] Francis, Richard L., "A Note on the Optimum Location of New Machines in Existing Plant Layouts," The Journal of Industrial Engineering *14*, 57–59 (1963).

[5] Francis, Richard L., "On the Location of Multiple New Facilities with Respect to Existing Facilities," The Journal of Industrial Engineering *15*, 106–107 (1964).

[6] Francis, Richard L., "On Some Problems of Rectangular Warehouse Design and Layout," The Journal of Industrial Engineering *18*, 595–604 (1967).

[7] Francis, Richard L., "Sufficient Conditions for Some Optimum Property Facility Designs," Operations Research *15*, 448–466 (1967).

[8] Kuhn, H. W. and R. E. Kuenne, "An Efficient Algorithm for the Numerical Solution of the Generalized Weber Problem in Spatial Economics," Journal of Regional Science *4*, 21–33 (1962).

[9] Love, R. F., "Locating Facilities In Three-Dimensional Space By Convex Programming," Nav. Res. Log. Quart. *16*, 503–516 (1969).

[10] Miehle, William, "A Link Length Minimization in Networks," Operations Research *6*, 232–243 (1958).

[11] Prager, W., "On the Design of Communication and Transportation Networks," Symposium on the Theory of Traffic Flow, Warren, Michigan, 1959.

[12] Rogers, J. D. and R. C. Vergin, "An Algorithm and Computational Procedure for Locating Economic Facilities," Management Science *13*, 240–254 (1967).

[13] Spang, H. A., "A Review of Minimization Techniques for Nonlinear Functions," Siam Review *4*, 343–365 (1962).

[14] Wesolowsky, G. O. and R. F. Love, "The Optimal Location of New Facilities Using Rectangular Distances," Operations Research *19*, 124–130 (1971).

A TECHNIQUE WHICH COMBINES MODIFIED PATTERN SEARCH METHODS WITH COMPOSITE DESIGNS AND POLYNOMIAL CONSTRAINTS TO SOLVE CONSTRAINED OPTIMIZATION PROBLEMS

B. F. Houston

and

R. A. Huffman

Vought Aeronautics Division
LTV Aerospace Corporation
Dallas, Texas

ABSTRACT

This paper presents a method of selecting design parameters which optimizes a specific measure (aircraft design example: minimum weight, maximum mission effectiveness) and guarantees designated levels of response in specified areas (such as combat ceiling, acceleration time). The method employs direct search optimization applied to a nonlinear functional constrained by nonlinear surfaces. The composite design technique is combined with regression methods to determine adequate surface representations with a minimum of required data points. A sensitivity analysis is conducted at the optimum set of design parameters to test for uniqueness.

1.0 INTRODUCTION

Constrained optimization problems are concerned with maximizing or minimizing some response, or functional value, subject to constraints on the system. The functional and constraints may or may not be linear.

The problem at hand is to minimize takeoff weight, defined as the functional value, and to determine other aircraft design parameters which meet a specific set of performance requirements. One approach to the problem may be to perform the experiment by systematic variation (factorial design) of all independent parameters while evaluating the functional at all points which satisfy the constraints. The optimum solution is then chosen as that set of experimental conditions which give a minimum functional value with all constraints satisfied. A second approach may be to first evaluate the response in order to locate a feasible set of parameter values. Subsequent responses are evaluated at perturbed parameter values so as to determine and move in a direction of the optimum functional value while satisfying all constraints (direct search methods). The third, and recommended, approach employs composite factorial designs. These designs permit the estimation of functional and constraint surfaces with a minimum number of experimental points. This leads to analytical surfaces to be used in direct search optimization procedures.

The last procedure requires conducting the experiment for a composite of a low level factorial design and using the resulting experimental data to estimate functional and constraint surfaces by means of second order polynomial fits. The tangent search optimization procedure is used to find the optimum set of conditions satisfying all constraints and minimizing the objective function. This optimum

set of conditions is then perturbed, one variable at a time, and upper and lower bounds are obtained on each variable in the neighborhood of the optimum. These bounds are determined such that none of the constraints are violated outside a small tolerance limit. Inferences may be made about which variables have an essentially unique optimum value and which ones satisfy the constraints at the optimum within some small interval. This indicates multiple solutions to the initial problem.

2.0 PROCEDURES

2.1 Design of Experiments

The problem is to maximize or minimize some response Y for a system in which prior experience indicates that independent variables $X_1, X_2, \ldots, X_k$ are the pertinent variables having the most significant effect on Y. It is desirable to obtain an adequate solution by employing the fewest number of actual experiments. The most conservative number of measurements at each X_i that shows any trend with respect to Y is two. To perform an experimental design to observe responses, a p^k factorial design can be conducted where k is the number of factors, i.e., independent variables, and p is the number of levels or particular value settings of each factor. If, for example, a conservative p^k experimental design is performed to observe responses affected by three independent variables, 2^3 design points would be selected. Table I indicates the treatment combinations at each design point where -1 and $+1$ represent the low and high levels of the ith independent variable.

TABLE I. 2^3 Design

Independent Variable Levels			Observed Response
X_1	X_2	X_3	Y
-1	-1	-1	Y_1
1	-1	-1	Y_2
-1	1	-1	Y_3
1	1	-1	Y_4
-1	-1	1	Y_5
1	-1	1	Y_6
-1	1	1	Y_7
1	1	1	Y_8

If it is unknown whether the response is affected by interactions between two or more X_i variables or by the second order effects of the X_i variables, then $2k+1$ points can be added to the 2^k design points already used to obtain a composite design. This permits the fitting of second order surfaces, Reference [1]. The $2k+1$ additional points consist of one at the center of the design and the remaining six in pairs along the coordinate axes at $\pm\alpha_1$, $\pm\alpha_2$, and $\pm\alpha_3$, respectively. The design matrix is given in Table II where composite points are added to those of Table I. Figure 1 is a diagram of the array of design points in three dimensional space.

If $\alpha_1 = \alpha_2 = \ldots = \alpha_k = \alpha$ for a suitably chosen α in a composite design for k factors, an orthogonal design can be constructed. Orthogonality permits response contributions due to main effects and interactions to be estimated independently. A manner in which orthogonal composite designs are built up for the number of factors, k, equal to 2, 3, 4, and 5 is shown in Table III. The second row of the table shows the number of design points for two level designs forming the nuclei of the composite designs. These two level designs are all complete factorials except the design for five factors, which

TABLE II. *A Three Factor Composite Design*

Trial	Factor Level		
	X_1	X_2	X_3
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	-1
5	-1	-1	1
6	1	-1	1
7	-1	1	1
8	1	1	1
9	$-\alpha_1$	0	0
10	α_1	0	0
11	0	$-\alpha_2$	0
12	0	α_2	0
13	0	0	$-\alpha_3$
14	0	0	α_3
15	0	0	0

is a half replicate of a complete 2^5 design. In the fourth row the values of α are given which make the corresponding composite designs orthogonal. Table III was obtained from Reference [2].

TABLE III. *Orthogonal Composite Designs*

Number of factors (k)	2	3	4	5
Basic two-level design points	2^2	2^3	2^4	$\frac{1}{2} 2^5$
Number of extra points ($2k + 1$)	5	7	9	11
Distance of axial points from center (α)	1.000	1.215	1.414	1.547

2.2 Polynomial Regression

Polynomial regression is especially useful and sometimes easier, when the response Y is thought to be affected by a large number of X variables and the theory behind a rigorously derived model for Y as a function of Xs is complex. As long as only the optimum set of conditions is desired, a polynomial model which fits the data well can lead to a set of optimal conditions approximating those of a rigorously derived model.

The data points from a p^k composite design may be used to fit the constraint equations as a function of the independent variables. The specific aircraft design problem treated here employs four independent variables and sixteen performance constraints. Thus a 2^4 composite design is used. For each of the resulting 25 data points, the desired aircraft performance parameters are measured or estimated by means of an experiment (in this case, a computer run). These performance values when coupled with the performance requirements, give rise to the constraint inequalities. That is, for the desired solution

$$\text{PERFORMANCE}_{\text{REQ}} \geq \text{PERFORMANCE MEAS.}$$

the constraint becomes

$$\text{CONSTRAINT} = \text{PERFORMANCE}_{\text{REQ}} - \text{PERFORMANCE}_{\text{MEAS}} \geq 0$$

and for the desired relation

$$\text{PERFORMANCE}_{\text{REQ}} \leq \text{PERFORMANCE}_{\text{MEAS}}$$

the constraint becomes

$$\text{CONSTRAINT} = \text{PERFORMANCE}_{\text{MEAS}} - \text{PERFORMANCE}_{\text{REQ}} \geq 0.$$

For each performance constraint and for the functional objective, there are 25 values corresponding to the 25 sets of independent parameter values. Response surfaces approximating these data can be derived by use of regression analysis. The step-up regression program used adds only first and second order terms in the order of greatest improvement in correlation between the actual and estimated response. (Estimated response is obtained from the polynomial equation.) Terms are added until the desired correlation is obtained.

2.3 Constrained Optimum Search Procedure

A mathematical statement of the optimizing procedure follows:

Let $X = (X_1, X_2, \ldots, X_k)$ be a vector of independent variables in k-dimensional space. Let $F(X)$ and $C_j(X)$, $(j = 1, 2, \ldots, m)$, be continuous functions which can be evaluated for any X within the vector space. Any X for which $C_j(X) \geq 0$, $(j = 1, 2, \ldots, m)$ will be called "feasible." The general problem of constrained minimization is to determine the vector X such that $F(X_*)$ is less than all other $F(X)$, subject to the constraints, $C_j(X_*) \geq 0$, for all $j = 1, 2, \ldots, m$. If for any point X, $C_j(X) \leq 0$, then the jth constraint is "violated" at X. Often $C_i(X_*) = 0$ for $1 < i < m$, indicating that the solution lies on one or more constraint boundaries.

The direct or pattern search method of Hooke and Jeeves [5] involves choosing a set of X_i values for the vector X and computing the functional to be optimized. A search is conducted with a sequence of base points until an optimum is eventually reached. Glass and Cooper's [3] sequential search method is a modification of the direct search of Hooke and Jeeves. The method continues in a successful direction, one of decreasing objective function if a minimum is required, by a sequence of base points, eliminating exploratory points about base points except when pattern moves fail. The Hillrary tangent search method employed here uses the Glass and Cooper modification of direct search except that whenever a move from a feasible base point violates a constraint, the exploratory procedure is performed in hyperplanes approximately tangent to the constraint hypersurfaces violated.

Only a very basic intuitive idea of the Hillrary search was presented above since Reference [4] can be used for details.

2.4 Constrained Optimization Using Composite Designs and Polynomial Regression

Composite designs and polynomial regression are employed to find adequate models for the constraint surfaces and functional or response to be optimized. The resulting polynomials are then used in constrained optimum search procedures, such as the one discussed in the preceding section, to find local optimum conditions. A sensitivity analysis is conducted to determine uniqueness of the optimum. This is illustrated in Section 3 of the present problem.

3.0 CONSTRAINED OPTIMIZATION OF AIRCRAFT TAKEOFF WEIGHT

3.1 The Problem

In the aircraft design at hand, takeoff weight, W_{TO}, aspect ratio, AR, wing surface area, S, and thrust, T, are independent variables, and it is desired to minimize takeoff weight under specified performance requirements.

The ranges of independent variables are:

(1) $$45000 \leq W_{TO} \leq 97000$$

(2) $$5 \leq AR \leq 9$$

(3) $$300 \leq S \leq 900$$

(4) $$30000 \leq T \leq 55000$$

The functional to be minimized is

(5) $$F = W_{TO}$$

subject to eight upper and lower bound constraints obtained for the four variables in equations (1)–(4). Additional constraints used in this problem are as follows:

(9), (11) $$70 \leq W_{TO}/S \leq 130$$

(10), (12) $$0.5 \leq T/W_{TO} \leq 1.3$$

(13) $$(W_{TO})_{est} \geq (W_{TO})_{req}$$

(14) $$(T_{acc})_{req} \geq (T_{acc})_{calc}$$

(15) $$(CC)_{req} \leq (CC)_{max}$$

(16) $$(N_z)_{req} \leq (N_z)_{max}$$

Constraints (9) through (12) are obtained from the double limit inequalities. Constraint (13) forces the variable W_{TO} to be greater than the computed requirements; generally the equality will be forced, but any slack available could possibly be used in additional cargo capacity. Numerical values given to the required acceleration time (T_{acc}), combat ceiling (CC), and equilibrium turn g (N_z) are 22 sec, 58,000 ft, and 2.0, respectively. All constraining inequalities have been rearranged such that an independent variables set satisfying

$$C(j) \geq 0 \qquad (j = 1, 2, \ldots, 16)$$

represents a feasible solution (a solution satisfying constraints).

Table IV presents values of the independent variables and the associated set of constraint values. The first 2^4 design points are those used in the initial conservative design; $2(4)+1$ design points have been added to complete the composite design.

3.2 Composite Design

A full factorial design was constructed using 2^4 points and other points were added to complete the composite design. Table III for four factors was used to obtain α for the additional points. Since the standardized values of the four independent variables between 0 and 1 were at levels 0.25 and 0.75 from the permissible range -1 to $+1$, $\alpha=1.414$ from Table 2 was adjusted to 0.3535 by the proportionality:

$$\frac{\alpha^1}{0.5}=\frac{1.414}{2} \qquad (5)$$

$$\alpha^1=0.3535$$

and normalized levels of each variable used in the design occurred at $0.5-\alpha^1=0.1465$, 0.5, and $0.5+\alpha^1=0.8535$ for the nine additional design points. Note that 0.5 is the center point coordinate of the design. Table IV shows constraint responses for all $2^4+2(4)+1=25$ treatment combinations of the composite design. Only raw, nonstandardized, values of the four independent variables are shown in the table.

TABLE IV. *Design Point Constraint Values For 2^4 Composite Design Experiment*

Design Point Number	Independent Variables				Constraint Values							
	$W_{TO}=x_1$	$AR=x_2$	$S=x_3$	$T=x_4$	$C(9)$	$C(10)$	$C(11)$	$C(12)$	$C(13)$	$C(14)$	$C(15)$	$C(16)$
1	50000	6	450	42500	41.11	0.15	18.89	0.45	−3934	0.1144	−32.36	−0.1311
2	50000	6	450	67500	41.11	0.65	18.89	−0.05	−3934	1.306	7165	−0.1311
3	50000	6	750	42500	−3.333	0.15	63.33	0.45	−7970	−0.3057	−178.4	1.084
4	50000	6	750	67500	−3.333	0.65	63.33	−0.05	−7970	1.125	7001	1.084
5	50000	8	450	42500	41.11	0.15	18.89	0.45	−3658	0.144	−174.1	−0.1491
6	50000	8	450	67500	41.11	0.65	18.89	−0.05	−3658	1.318	7006	−0.1491
7	50000	8	750	42500	−3.333	0.15	63.33	0.45	−8432	−0.265	−299.1	1.059
8	50000	8	750	67500	−3.333	0.65	63.33	−0.05	−8432	1.143	6866	1.059
9	60000	6	450	42500	63.33	0.008333	−3.333	0.5917	2026	−0.5656	−2727	−0.4516
10	60000	6	450	67500	63.33	0.425	−3.333	0.1750	2026	1.015	4136	−0.4516
11	60000	6	750	42500	10.0	0.008333	50.0	0.5917	−1582	−1.129	−2884	0.5517
12	60000	6	750	67500	10.0	0.425	50.0	0.1750	−1582	7723	3960	0.5517
13	60000	8	450	42500	63.33	0.008333	−3.333	0.5917	2617	−0.5286	−2870	−0.4674
14	60000	8	450	67500	63.33	0.425	−3.333	0.1750	2617	1.030	3970	−0.4674
15	60000	8	750	42500	10.0	0.008333	50.0	0.5917	−1786	−1.077	[illegible]	0.5299
16	60000	8	750	67500	10.0	0.4250	50.0	0.1750	−1786	0.7947	3825	0.5299
17	47930	7	600	55000	9.883	0.4475	50.12	0.1525	−7181	0.8598	4398	0.5761
18	62070	7	600	55000	33.45	0.1861	26.55	0.4139	1830	0.1957	302.3	−0.0301
19	55000	5.586	600	55000	21.67	0.3000	38.33	0.3000	−2757	0.5192	2289	0.2487
20	55000	8.414	600	55000	21.67	0.3000	38.33	0.3000	−2700	0.5550	2092	0.2198
21	55000	7	387.9	55000	71.79	0.3000	−11.79	0.3000	13.57	0.7814	2320	−0.5422
22	55000	7	813.9	55000	−2.424	0.3000	62.42	0.3000	−5903	0.3482	2097	1.012
23	55000	7	600	37320	21.67	−0.02136	38.33	0.6214	−2614	−1.178	−3447	0.2320
24	55000	7	600	72670	21.67	0.6214	38.33	−0.02136	−2641	1.202	6570	0.2320
25	55000	7	600	55000	21.67	0.3000	38.33	0.3000	−2641	0.5382	2175	0.2320

3.3 Polynomial Regression

A step-up regression program was used to fit the data of Table IV to second order polynomial models. The first eight constraints $C(1)$ -- $C(8)$ were merely upper and lower bounds on W_{TO}, AR, S, and, T as obtained from Equations (1)—(4). For each regression, terms were added one-at-a-time to the model until the multiple correlation coefficient R reached 0.999. This means that a plot of observed responses against calculated responses would give a line very close to a 45 degree line passing through the origin and lying in the first quadrant: this implies a close estimate of calculated to observed responses. The regression constraint surfaces are shown in Table V.

TABLE V. *Quadratic Regression Models For 2^4 Composite Design Constraint Surfaces*

Constraint Equation
$C(9)=26.8641+0.003538X_1-0.3375X_3+0.2796E-3\times X_3^2-0.297E-5\times X_1X_3$
$C(10)=-0.7001-0.367E-4\times X_1-0.3341E-9\times X_1X_1$
$C(11)=33.5015-0.003546X_1+0.337X_3-0.2798E-3\times X_3^2+0.2983E-5\times X_1X_3$
$C(12)=1.30016-0.3669E-4\times X_1+0.3342E-9\times X_1X_1$
$C(13)=-64906.5231+1.8968X_1-10.5779X_2-0.1298E-4\times X_1^2+0.01428\times X_1X_2+0.1067E-3\times X_1X_4-1.2702X_2X_3$
$C(14)=1.0815-0.1261E-3\times X_1-0.001286X_4+0.1233E-3\times X_1-0.162E-8\times X_1^2+0.3424E-7\times X_1X_3+0.1719E-8\times X_1X_1+0.3712E-7\times X_3X_1$
$C(15)=-3305.6797-0.2872\times X_1+0.4960X_1-0.1954E-5\times X_1^2$
$C(16)=-1.9586+0.007581X_3-0.7091E-7\times X_1X_3$

3.4 Constrained Optimization Search Procedure and Sensitivity Analysis

The Hilleary tangent search program, Reference [4], was employed to minimize the takeoff weight subject to the bound equations, Equations (1)—(4), and the constraints of Table V. The search terminal values at the optimum are recorded in the second column of Table VI. A short program was written to perturb these optimum values slightly one-at-a-time until none of the constraint values went below a selected tolerance limit, -0.005. Recall that negative constraint values constitute a violation. Upper and lower limits were obtained on the optimum for each variable while holding all other variables constant. The percentage variations of upper from lower limits are indicated in column five of the table. The variations for takeoff weight W_{TO} and wing surface area S are both less than one percent, so it may be assumed that the optimum values for these variables are essentially unique (constant).

The values of aspect ratio AR and thrust T at the optimum are not unique, that is, any value of aspect ratio between the upper and lower limits may be selected and similarly for thrust. Table V equations indicate no interaction between surface area and thrust, X_3X_4 terms, so the preceding statement of freedom of choice for both variables within the Table VI limits is verified. If an X_3X_4 term

TABLE VI. *Optimum Solution And Limits*

Variable at Optimum	Search Terminal Value	Lower Limit	Upper Limit	Percent Variation*
Takeoff Weight, W_{TO}	58.157	58.157	58.282	0.215
Aspect Ratio, AR	8.828	8.828	8.928	1.132
Wing Surface Area, S	566.90	565.96	566.90	0.166
Thrust	61.100	50.310	75.650	50.3

$$\text{* Percent Variation}=\frac{(\text{Upper Limit}-\text{Lower Limit})}{\text{Lower Limit}}\times 100.$$

were present in the constraint equations, then different aspect ratios could be plotted against thrust upper and lower bounds to determine the range of choice for thrust at a selected aspect ratio. In actual aircraft design it is unlikely that the thrust parameter will have such latitude, but it is probable that some of the independent variables will not be effective and hence not controlled.

4.0 CONCLUSIONS

If constrained (or unconstrained) optimization techniques are to be applied to empirical data obtained from costly experiments, analysis costs may be greatly reduced by employing the procedures outlined herein. This involves the combination of a composite experimental design to minimize the number of required experiments with regression analysis to obtain analytical approximations of the experimental data. Finally classical constrained optimization techniques are used with data from the analytical form of the response surface. The resulting optimum solution will, because of regression, approximate the optimum which could be obtained using actual experimental data. It is also desirable to conduct a sensitivity analysis about the optimum to investigate the uniqueness of the solution. The response surface model would again give approximate results while actual experimental data would give exact results.

REFERENCES

[1] Cochran, W. G., and G. M. Cox, *Experimental Designs* (John Wiley & Sons, Inc., New York, 1964).

[2] Davies, O. L., *Design and Analysis of Industrial Experiments* (Oliver & Boyd, Edinburgh, 1954).

[3] Glass, H., and L. Cooper, "Sequential Search: A Method for Solving Constrained Optimization Problems," J. Assoc. for Computing Machinery, Vol. 12 (Jan. 1965).

[4] Hilleary, R. R., *The Tangent Search Method of Constrained Minimization*, United States Naval Postgraduate School, Technical Report/Research Paper No. 59 (Mar. 1966).

[5] Hooke, R., and F. A. Jeeves, "'Direct Search' Solution of Numerical and Statistical Problems," J. Assoc. for Computing Machinery, Vol. 8, No. 2 (Apr. 1961).

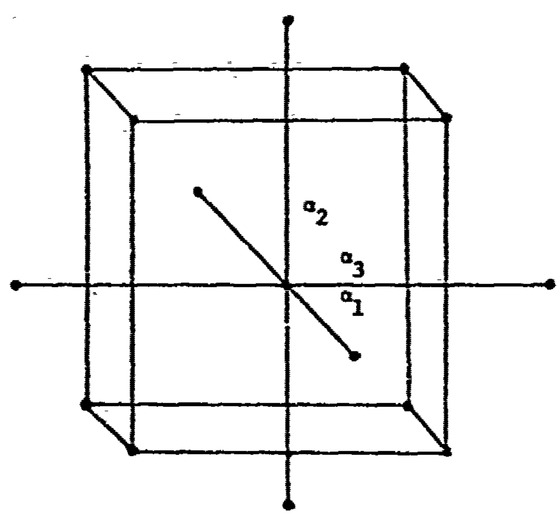


FIGURE 1. Three-factor composite design array

FINITE STATISTICAL GAMES AND LINEAR PROGRAMMING

Robert A. Agnew

Air Force Institute of Technology

and

Roy B. Hempley

Headquarters USAF

ABSTRACT

The dual linear programs associated with finite statistical games are investigated and their optimal solutions are interpreted. The usual statistical game is generalized to a two-sided (inference) game and its possible application as a tactical model is discussed.

1. INTRODUCTION

In this paper, we investigate the dual linear programs associated with finite statistical games (i.e., finite strategy and observation spaces) and interpret their optimal solutions. Furthermore, we generalize the usual (one-sided inference) statistical game to a two-sided (inference) statistical game where each player makes an initial strategy choice, partially implements his initial strategy while making an observation allowing him to infer about his opponent's initial choice, and finally makes a secondary choice restricted by the partial implementation of his initial strategy.

In Sections 2, 3, and 4, we present a hierarchy of three games and their associated dual linear programs. Each game in the hierarchy contains its predecessor as an imbedded special case. Section 2 treats the familiar Rectangular Game. Section 3 treats the Statistical Game of Statistical Decision Theory. Section 4 generalizes to the Two-Sided Statistical Game. Section 5 briefly discusses the Two-Sided Statistical Game as a tactical warfare model. All summations are over their full ranges unless otherwise specified.

2. RECTANGULAR GAME

Suppose we have two players Blue and Red, with finite pure strategy sets indexed over I and J, respectively. Let a_{ij} be the payoff to Blue when he chooses pure strategy $i \epsilon I$ and Red chooses pure strategy $j \epsilon J$; we make the zero-sum assumption that $-a_{ij}$ is the corresponding payoff to Red. Let x_i denote the probability that Blue chooses pure strategy i, and let y_j be the probability that Red chooses pure strategy j. The Minimax Theorem assures the existence of a saddle point in mixed strategies. Moreover, we have the familiar dual linear programs

$$\max v$$

$$\text{(1)} \qquad \text{s.t.} \sum_i a_{ij} x_i \geq v; \quad j \epsilon J$$

$$\sum_i x_i = 1$$

$$x_i \geq 0; \quad i \epsilon I$$

$$\min u$$

$$\text{(2)} \qquad \text{s.t.} \sum_j a_{ij} y_j \leq u: \quad i \in I$$
$$\sum_j y_j = 1$$
$$y_j \geq 0: \quad j \in J$$

either of which can be solved to yield optimal strategies and the value of the game.

3. STATISTICAL GAME

Suppose that Red chooses first and that Blue makes one of a finite number of mutually exclusive and exhaustive observations indexed over K, thereby allowing him to infer about Red's choice before making his choice. Let p_{jk} denote the probability (assumed known) that Blue observes $k \in K$ given that Red has chosen pure strategy j. This game can obviously be formulated as a rectangular game where Blue has pure strategies of the form $\{i_k\}_{k \in K}$ implying his choice of $i_k \in I$ after observing k. The expected payoff to Blue is $\sum_k a_{i_k j} p_{jk}$ when he chooses pure strategy $\{i_k\}_{k \in K}$ and Red chooses pure strategy j. One can write down the analogs to (1) and (2) for this game, but trivial rearrangements and substitutions yield the equivalent dual linear programs

$$\max r$$

$$\text{(3)} \qquad \text{s.t.} \sum_k \sum_i a_{ij} p_{jk} x_{ki} \geq r: \quad j \in J$$
$$\sum_i x_{ki} = 1: \quad k \in K$$
$$x_{ki} \geq 0: \quad i \in I,\ k \in K$$

$$\min \sum_k u_k$$

$$\text{(4)} \qquad \text{s.t.} \sum_j a_{ij} p_{jk} y_j \leq u_k: \quad i \in I,\ k \in K$$
$$\sum_j y_j = 1$$
$$y_j \geq 0: \quad j \in J$$

Suppose we denote the optimal solutions to (3) and (4) with stars. The x_{ki}^* represent Blue's optimal mixed strategy in behavioral form: i.e., x_{ki}^* is the probability that Blue chooses i after observing k. The optimal dual variables yield a nice interpretation for Blue. u_k^* represents a partial value of the game with respect to observation k, and $\sum_j y_j^* p_{jk}$ is the probability that k is observed. Hence, we can interpret $u_k^* / \sum_j y_j^* p_{jk}$ as the conditional (interim) value of the game to Blue given that he has observed k.

In conventional statistical decision theory, Blue represents the statistician and Red represents nature. A mixed strategy for Blue is called a "randomized decision rule" and his optimal strategy is called "maximin." A mixed strategy for Red is called an "a priori distribution" and his optimal strategy is called "least favorable." We remark that there exist situations [9] where the payoff depends on the observation (i.e., a_{ijk}). It should be clear that this generalization can be handled identically.

4. TWO-SIDED STATISTICAL GAME

We now assume that each player makes an initial choice, makes an observation allowing him to infer about his opponent's choice, and then makes a secondary choice from a restricted set of pure

strategies. We use the previous notation for Blue, and we denote his admissible secondary strategy set by $M_i \subset I$ after initial choice i; we assume $i \epsilon M_i$. We assume that Red makes one of a finite number of mutually exclusive and exhaustive observations indexed over T, and we let q_{it} denote the probability that Red observes $t \epsilon T$ given that Blue has chosen i initially. We denote Red's admissible secondary strategy set by $N_j \subset J$ after initial choice j; we assume that $j \epsilon N_j$. This game can be formulated as a rectangular game with pure strategies of the form $(i, \{i_k\}_{k \epsilon K})$ with $i_k \epsilon M_i$ and $(j, \{j_t\}_{t \epsilon T})$ with $j_t \epsilon N_j$ for Blue and Red, respectively. The expected payoff to Blue is $\sum_k \sum_t a_{i_k j_t} p_{jk} q_{it}$ when he chooses $(i, \{i_k\}_{k \epsilon K})$ and Red chooses $(j, \{j_t\}_{t \epsilon T})$. The analogs to (1) and (2) reduce to the following dual linear programs.

max v

(5)
$$\text{s.t.} \sum_i \sum_k \sum_{m \epsilon M_i} a_{mn} p_{jk} q_{it} x_{ikm} \geq v_{jt};\ n \epsilon N_j,\ t \epsilon T,\ j \epsilon J$$

$$\sum_t v_{jt} = v;\ j \epsilon J$$

$$\sum_{m \epsilon M_i} x_{ikm} = w_i;\ k \epsilon K,\ i \epsilon I$$

$$\sum_i w_i = 1$$

$$x_{ikm} \geq 0;\ m \epsilon M_i,\ k \epsilon K,\ i \epsilon I$$

min u

(6)
$$\text{s.t.} \sum_j \sum_t \sum_{n \epsilon N_j} a_{mn} p_{jk} q_{it} y_{jtn} \leq u_{ik};\ m \epsilon M_i,\ k \epsilon K,\ i \epsilon I$$

$$\sum_k u_{ik} = u;\ i \epsilon I$$

$$\sum_{n \epsilon N_j} y_{jtn} = z_j;\ t \epsilon T,\ j \epsilon J$$

$$\sum_j z_j = 1$$

$$y_{jtn} \geq 0;\ n \epsilon N_j,\ t \epsilon T,\ j \epsilon J$$

Again, we have a behavioral representation of the optimal strategies. w_i^* is the probability that Blue chooses i initially, and if $w_i^* > 0$, x_{ikm}^* / w_i^* is his probability of subsequently choosing $m \epsilon M_i$ given that i was chosen initially and k was observed. Again, the optimal dual variables yield a nice interpretation for Blue. We can interpret $u_{ik}^* / \sum_j z_j^* p_{jk}$ as the conditional value of the game to Blue given that he chose i initially and observed k.

5. TACTICAL APPLICATION

The Two-Sided Statistical Game may be useful as a model for tactical warfare where both sides have reconnaissance capabilities. Suppose that a_{ij} represents Blue's subjective probability of mission accomplishment given that he selects i and Red selects j. The value of the game is then Blue's unconditional subjective probability of mission accomplishment. Zero-sum payoff represents the usual conservative "worst case" criterion.

A value can be assigned to any particular type of reconnaissance (e.g., aerial) by solving the game with and without it, the difference in values representing the increment of subjective probability of mission accomplishment contributed by the specified type of reconnaissance.

REFERENCES

[1] Blackwell, D. and M. A. Girshick, *Theory of Games and Statistical Decisions* (John Wiley and Sons, Inc., N.Y., 1954).

[2] Charnes, A. and W. W. Cooper, *Management Models and Industrial Applications of Linear Programming* (John Wiley and Sons, Inc., N.Y., 1961), Vol. II.

[3] Dantzig, G. B., *Linear Programming and Extensions* (Princeton University Press, Princeton, N.J., 1963).

[4] Dresher, M., *Games of Strategy: Theory and Application* (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1961).

[5] Feller, W., *An Introduction to Probability Theory and Its Applications* (John Wiley and Sons, Inc., N.Y., Vol. I, 3rd. Ed., 1968).

[6] Ferguson, T. S., *Mathematical Statistics: A Decision Theoretic Approach* (Academic Press, N.Y., 1967).

[7] Hempley, R. B., "Game Theoretic Models for a Limited Tactical Warfare Engagement," Unpublished M. S. Thesis, Air Force Institute of Technology, 1969.

[8] Owen, G., *Game Theory* (W. B. Saunders Co., Philadelphia, 1968).

[9] Weiss, L., *Statistical Decision Theory* (McGraw-Hill, N.Y., 1961).

ON QUEUES WITH STATE-DEPENDENT ERLANG SERVICE

Carl M. Harris

The George Washington University
School of Engineering and Applied Science
Institute for Management Science and Engineering

ABSTRACT

Some general results are derived for single-channel queues with Poisson input and state-dependent Erlang service times in view of the possible use of this model to approximate arbitrary $M/G/1$-like state-dependent queues in a manner similar to that suggested by Rosenshine, and by Kendall, and Kotiah, Thompson, and Waugh for the $M/G/1$. Numerical procedures are indicated for the evaluation of stationary state probabilities, expected system sizes and waiting times, and parameter estimation.

1. INTRODUCTION

In view of the history of the use of Erlangian distributions as approximations for more general types of inter-arrival and service-time distributions in single-server systems (see Kendall [5]) and of recent papers by Rosenshine [7], and Kotiah, Thompson, and Waugh [6], some results are developed in this paper for queues with Poisson input, state-dependent Erlang service, and one server. In view of the reasonably favorable results the authors of [6] found in their evaluation of Erlangian approximations for $GI/G/1$, it is suggested here that a similar approach could be used for queues with Poisson input and general but state-dependent service times and that these models should provide reasonable estimates. The particular emphasis of this paper is upon the procedure for determining stationary probabilities for the $M/G/1$ state-dependent queue with Erlang service.

The general state-dependent system with Poisson input can be described in the following manner:

(1) Customers arrive as a simple Poisson process with parameter λ.

(2) The customers are serviced singly, first come, first served, by one server.

(3) The service time of each customer is conditioned on the number in the queue. Service times of customers beginning service with the same number n in the system are independent and identically distributed random variables, $\{T_n, n=1, 2, \ldots\}$, with *cdf*

$$B_n(t) = Pr\{T_n \leq t\}.$$

The departure process of this system, $\{X_k, k=1, 2, \ldots | X_k = \text{number of customers in the system immediately after the } k\text{th customer leaves the system}\}$, is a Markov chain, independent of the form of the set of *cdf*'s $\{B_n(t)\}$ [4]. The transition matrix, $P=[p_{ij}]$, of this imbedded chain is given by

$$\begin{bmatrix} k_{01} & k_{11} & k_{21} & k_{31} & \cdot & \cdot & \cdot \\ k_{01} & k_{11} & k_{21} & k_{31} & \cdot & \cdot & \cdot \\ 0 & k_{02} & k_{12} & k_{22} & \cdot & \cdot & \cdot \\ 0 & 0 & k_{03} & k_{13} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

where

(1) $k_{ij}=Pr\{i$ arrivals during a full service period $|\, j$ were in the system when service began$\}$

$$=(1/i!)\int_0^\infty e^{-\lambda t}(\lambda t)^i dB_j(t).$$

It is clearly seen for $i>0$ that

$$p_{ij}=k_{j-i+1,i}\quad (j\geq i-1),$$

and that

$$p_{0j}=k_{j1}.$$

II. RESULTS

More specifically now, it will be assumed that

$$dB_n(t)=\frac{\mu_n(\mu_n t)^{s_n-1}e^{-\mu_n t}}{(s_n-1)!}dt\quad \text{(for all } n\text{)},$$

that is, that the service times of n-type customers are distributed according to an Erlang type $-s_n$. Hence, from (1),

$$k_{ij}=(1/i!)\int_0^\infty e^{-\lambda t}(\lambda t)^i\frac{\mu_j(\mu_j t)^{s_j-1}}{(s_j-1)!}e^{-\mu_j t}dt$$

$$=\frac{\lambda^i\mu_j^{s_j}}{i!(s_j-1)!}\int_0^\infty e^{-(\lambda+\mu_j)t}t^{i+s_j-1}dt$$

$$=\frac{\lambda^i\mu_j^{s_j}}{i!(s_j-1)!}\frac{(i+s_j-1)!}{(\lambda+\mu_j)^{i+s_j}}$$

$$=\binom{i+s_j-1}{i}(\lambda/\mu_j)^i/(\lambda/\mu_j+1)^{i+s_j}.$$

If the utilization factor ρ_j is defined as

$$\rho_j=s_j\lambda/\mu_j,$$

then

(2) $$k_{ij}=\binom{i+s_j-1}{i}(\rho_j/s_j)^i/(\rho_j/s_j+1)^{i+s_j}.$$

The chain is ergodic when

$$\lim\sup_{j\to\infty}\{\rho_j=\sum_{i=1}^\infty ik_{ij}=s_j\lambda/\mu_j\}<1\quad [5].$$

When the steady-state distribution, $\{\pi_i\}$, of the system size exists, it is the solution of the system

$$\sum_{i=0}^\infty p_{ij}\pi_i=\pi_j\qquad (j=0,1,2,\ldots).$$

and

$$\sum_{i=0}^\infty \pi_i=1.$$

From the transition matrix,

$$\text{(3)} \qquad \begin{aligned} \pi_j &= \sum_{i=0}^{\infty} p_{ij}\pi_i \\ &= k_{j1}(\pi_0 + \pi_1) + \sum_{i=2}^{j+1} k_{j-i+1,i}\,\pi_i. \end{aligned}$$

If (3) is multiplied by z^j and then summed on j, it is found that

$$\begin{aligned} \sum_{j=0}^{\infty} \pi_j z^j = (\pi_0 + \pi_1) \sum_{j=0}^{\infty} k_{j1} z^j \\ + \sum_{j=1}^{\infty} \sum_{i=2}^{j+1} k_{j-i+1,i}\,\pi_i z^j. \end{aligned}$$

If the generating functions are defined as

$$\Pi(z) = \sum_{j=0}^{\infty} \pi_j z^j$$

and

$$K_i(z) = \sum_{j=0}^{\infty} k_{ji} z^j \qquad (i = 0, 1, 2, \ldots),$$

then the aforementioned becomes

$$\text{(4)} \qquad \Pi(z) = \pi_0 K_1(z) + (1/z)[\pi_1 z K_1(z) + \pi_2 z^2 K_2(z) + \pi_3 z^3 K_3(z) + \ldots].$$

It should also be pointed out that all results for the imbedded chain can be directly related to the general-time process, $\{X(t), t\epsilon(0, \infty)|X(t) =$ number of customers in the system at time $t\}$, since the stationary probabilities of the chain are identical to those of $X(t)$, namely p_n. A proof of this assertion for the single-server Poisson queue with general but state-dependent service times is contained in [3].

As an illustration of this approach, let the service distribution of those alone in the system be a different Erlang from that governing all other customers. That is,

$$b(t) = dB_n(t)/dt = \begin{cases} \mu_1(\mu_1 t)^{s_1-1}e^{-\mu_1 t}/(s_1-1)! & (n = 1) \\ \mu(\mu t)^{s-1}e^{-\mu t}/(s-1)! & (n > 1). \end{cases}$$

Hence, from (2),

$$\text{(5)} \qquad \begin{aligned} k_{i1} &= Pr\{i \text{ arrivals during service time}|1 \textit{ in system when service began}\} \\ &= \binom{i + s_1 - 1}{i} (\rho_1/s_1)^i/(\rho_1/s_1 + 1)^{i+s_1} \end{aligned}$$

and

$$\text{(6)} \qquad k_i = \binom{i + s - 1}{i} (\rho/s)^i/(\rho/s + 1)^{i+s}.$$

The transition matrix P is therefore given by

$$\begin{bmatrix} k_{01} & k_{11} & k_{21} & k_{31} & \cdot & \cdot & \cdot \\ k_{01} & k_{11} & k_{21} & k_{31} & \cdot & \cdot & \cdot \\ 0 & k_0 & k_1 & k_2 & \cdot & \cdot & \cdot \\ 0 & 0 & k_0 & k_1 & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & k_0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

and the generating functions by

$$K_1(z)=\sum_{j=0}^{\infty} k_{j1}z^j$$

and

$$K_i(z)=K(z)=\sum_{j=0}^{\infty} k_j z^j \qquad (i=2,3,\ldots),$$

where k_{j1} and k_j would be given by (5) and (6), respectively. Therefore, from Equation (4),

(7) $$\Pi(z)=\{[zK_1(z)-K(z)]\pi_0+z[K_1(z)-K(z)]\pi_1\}/[z-K(z)].$$

π_0 and π_1 are obtained by first finding the limit of $\Pi(z)$ as z goes to one. Since $K_i(1)=1$, l'Hôpital's rule is used, and it is found that

$$\lim_{z\to 1}\Pi(z)=\{[K_1(1)+K_1'(1)-K'(1)]\pi_0+[K_1(1)-K(1)+K_1'(1)-K'(1)]\pi_1\}/[1-K'(1)].$$

But

$$K_i'(1)=\sum_{j=1}^{\infty} jk_{ji}=\rho_i.$$

Hence

$$\lim_{z\to 1}\Pi(z)=1=\pi_0(1+\rho_1-\rho)/(1-\rho)+\pi_1(\rho_1-\rho)/(1-\rho),$$

where

$$\rho_i=\rho \text{ for } i\geq 2.$$

A second equation in π_0 and π_1 is obtained from (3), namely,

$$\pi_0=k_{01}(\pi_0+\pi_1).$$

When these two linear equations in π_0 and π_1 are solved simultaneously, it is found that

$$\pi_0=k_{01}(1-\rho)/(k_{01}+\rho_1-\rho)$$

and

$$\pi_1=(1-k_{01})(1-\rho)/(k_{01}+\rho_1-\rho),$$

where

$$\rho_1=s_1\lambda/\mu_1$$

and

$$\rho=s\lambda/\mu.$$

But, from (5),

$$k_{01}=1/(\rho_1/s_1+1)^{s_1}.$$

Hence

$$\pi_0=[1-\rho]/[1+(\rho_1-\rho)(\rho_1/s_1+1)^{s_1}]$$

and

$$\pi_1 = [(\rho_1/s_1 + 1)^{s_1} - 1][1 - \rho]/[1 + (\rho_1 - \rho)(\rho_1/s_1 + 1)^{s_1}].$$

The remaining π_i would be obtained by iterating Equation (3). The stationary distribution would, in fact, exist whenever $\rho < 1$. Of course, the computation of the $\{\pi_i\}$ becomes rather cumbersome very quickly, so an appeal is made to the computer. Numerical values of the $\{\pi_i\}$ are displayed in Table 1 following the text for some typical values of ρ and ρ_1, with s_1 and s fixed at the not unusual values of 2 and 4, respectively. Of course, similar calculations can also be performed simply for arbitrary values of s_1 and s.

TABLE 1. *PI(I) vs RHO, RHO1 FIXED*

					RHO1 = 0.5				
RHO =	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
PI(0)	0.5538	0.5417	0.5333	0.5189	0.5000	0.4741	0.4361	0.3765	0.2667
PI(1)	0.3115	0.3064	0.3000	0.2919	0.2813	0.2667	0.2455	0.2118	0.1500
PI(2)	0.0993	0.1076	0.1157	0.1235	0.1301	0.1347	0.1352	0.1269	0.0976
PI(3)	0.0266	0.0308	0.0368	0.0447	0.0545	0.0659	0.0773	0.0848	0.0760
PI(4)	0.0066	0.0080	0.0104	0.0147	0.0214	0.0314	0.0447	0.0591	0.0632
PI(5)	0.0016	0.0019	0.0028	0.0045	0.0081	0.0147	0.0258	0.0416	0.0533
PI(6)	0.0004	0.0005	0.0007	0.0013	0.0030	0.0068	0.0149	0.0293	0.0451
PI(7)	0.0001	0.0001	0.0002	0.0004	0.0011	0.0031	0.0086	0.0207	0.0381
PI(8)	0.0000	0.0000	0.0000	0.0001	0.0004	0.0014	0.0050	0.0146	0.0323
PI(9)	0.0000	0.0000	0.0000	0.0000	0.0001	0.0007	0.0029	0.0103	0.0273
PI(10)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003	0.0016	0.0072	0.0231
PI(11)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0009	0.0051	0.0196
PI(12)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0005	0.0036	0.0166
PI(13)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003	0.0025	0.0140
PI(14)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0018	0.0119
PI(15)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0013	0.0100
L =	0.627	0.659	0.703	0.766	0.861	1.018	1.313	1.993	4.446
					RHO1 = 1.0				
PI(0)	0.2975	0.2857	0.2718	0.2553	0.2353	0.2105	0.1791	0.1379	0.0816
PI(1)	0.3719	0.3571	0.3398	0.3191	0.2941	0.2632	0.2239	0.1724	0.1020
PI(2)	0.1916	0.2026	0.2118	0.2181	0.2199	0.2148	0.1991	0.1668	0.1072
PI(3)	0.0833	0.0919	0.1027	0.1154	0.1288	0.1409	0.1472	0.1394	0.1014
PI(4)	0.0340	0.0381	0.0444	0.0535	0.0661	0.0818	0.0981	0.1074	0.0905
PI(5)	0.0131	0.0151	0.0181	0.0230	0.0313	0.0441	0.0616	0.0793	0.0781
PI(6)	0.0052	0.0058	0.0071	0.0094	0.0141	0.0227	0.0374	0.0572	0.0670
PI(7)	0.0020	0.0022	0.0027	0.0037	0.0061	0.0114	0.0223	0.0408	0.0570
PI(8)	0.0007	0.0008	0.0010	0.0014	0.0026	0.0055	0.0131	0.0290	0.0483
PI(9)	0.0003	0.0003	0.0004	0.0005	0.0011	0.0027	0.0076	0.0205	0.0409

TABLE 1. *PI(I) vs RHO, RHO1 FIXED*—Continued

RHO =	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
				RHO1 = 1.0					
PI(10)	0.0001	0.0001	0.0001	0.0002	0.0004	0.0013	0.0044	0.0145	0.0346
PI(11)	0.0000	0.0000	0.0000	0.0001	0.0002	0.0006	0.0026	0.0102	0.0293
PI(12)	0.0000	0.0000	0.0000	0.0000	0.0001	0.0003	0.0015	0.0072	0.0248
PI(13)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0009	0.0051	0.0210
PI(14)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0005	0.0036	0.0178
PI(15)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003	0.0025	0.0150
L =	1.262	1.327	1.414	1.533	1.705	1.973	2.436	3.395	6.397
				RHO1 = 2.0					
PI(0)	0.1047	0.0976	0.0897	0.0811	0.0714	0.0606	0.0484	0.0345	0.0185
PI(1)	0.3140	0.2927	0.2692	0.2432	0.2143	0.1818	0.1452	0.1034	0.0556
PI(2)	0.2310	0.2372	0.2397	0.2374	0.2288	0.2120	0.1845	0.1430	0.0834
PI(3)	0.1458	0.1542	0.1633	0.1722	0.1790	0.1807	0.1725	0.1476	0.0953
PI(4)	0.0876	0.0934	0.1009	0.1106	0.1217	0.1327	0.1391	0.1321	0.0955
PI(5)	0.0511	0.0546	0.0596	0.0668	0.0768	0.0896	0.1029	0.1093	0.0894
PI(6)	0.0292	0.0313	0.0343	0.0389	0.0462	0.0573	0.0722	0.0862	0.0805
PI(7)	0.0164	0.0176	0.0194	0.0222	0.0270	0.0353	0.0487	0.0658	0.0708
PI(8)	0.0091	0.0098	0.0108	0.0124	0.0154	0.0212	0.0320	0.0492	0.0614
PI(9)	0.0050	0.0054	0.0059	0.0069	0.0087	0.0124	0.0206	0.0362	0.0528
PI(10)	0.0027	0.0029	0.0032	0.0038	0.0048	0.0072	0.0130	0.0263	0.0451
PI(11)	0.0015	0.0016	0.0018	0.0021	0.0026	0.0041	0.0081	0.0190	0.0384
PI(12)	0.0008	0.0009	0.0009	0.0011	0.0014	0.0023	0.0050	0.0135	0.0326
PI(13)	0.0004	0.0005	0.0005	0.0006	0.0008	0.0013	0.0031	0.0097	0.0277
PI(14)	0.0002	0.0002	0.0003	0.0003	0.0004	0.0007	0.0019	0.0069	0.0233
PI(15)	0.0001	0.0001	0.0001	0.0002	0.0002	0.0004	0.0011	0.0049	0.0199
L =	2.293	2.384	2.499	2.652	2.865	3.182	3.704	4.747	7.852
				RHO1 = 5.0					
PI(0)	0.0147	0.0134	0.0120	0.0105	0.0089	0.0073	0.0056	0.0038	0.0020
PI(1)	0.1659	0.1505	0.1344	0.1177	0.1002	0.0820	0.0629	0.0429	0.0220
PI(2)	0.1599	0.1597	0.1567	0.1504	0.1402	0.1252	0.1046	0.0777	0.0432
PI(3)	0.1360	0.1388	0.1412	0.1421	0.1404	0.1341	0.1208	0.0972	0.0588
PI(4)	0.1121	0.1150	0.1182	0.1216	0.1243	0.1247	0.1196	0.1036	0.0682
PI(5)	0.0908	0.0932	0.0962	0.0999	0.1042	0.1081	0.1091	0.1013	0.0726
PI(6)	0.0725	0.0744	0.0770	0.0803	0.0847	0.0900	0.0948	0.0938	0.0732
PI(7)	0.0572	0.0588	0.0608	0.0637	0.0676	0.0731	0.0797	0.0837	0.0712
PI(8)	0.0438	0.0460	0.0477	0.0499	0.0533	0.0583	0.0654	0.0727	0.0671

TABLE I. $PI(\bar{I})$ *vs RHO, RHO1 FIXED*—Continued

$RHO1 = 5.0$—Continued									
$RHO=$	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
$PI(9)$	0.0348	0.0357	0.0370	0.0389	0.0416	0.0459	0.0528	0.0618	0.0626
$PI(10)$	0.0268	0.0276	0.0286	0.0300	0.0322	0.0358	0.0420	0.0517	0.0572
$PI(11)$	0.0206	0.0212	0.0219	0.0231	0.0248	0.0277	0.0331	0.0427	0.0517
$PI(12)$	0.0157	0.0162	0.0168	0.0176	0.0190	0.0213	0.0258	0.0348	0.0462
$PI(13)$	0.0119	0.0123	0.0128	0.0134	0.0145	0.0163	0.0200	0.0282	0.0410
$PI(14)$	0.0091	0.0093	0.0097	0.0102	0.0110	0.0124	0.0154	0.0226	0.0361
$PI(15)$	0.0068	0.0070	0.0073	0.0077	0.0083	0.0094	0.0118	0.0180	0.0316
$\bar{I} =$	4.752	4.819	4.975	5.134	5.359	5.682	6.216	7.270	10.366

Two interesting side computations fall out nicely from Equation (7), namely the expected system size and waiting time, since

$$E[N] = L = \Pi'(1).$$

L is found by the successive application of l'Hôpital's rule and use of the facts that

$$K_i(1) = 1$$

and

$$K_i'(1) = \rho_i,$$

and the expected waiting time W follows from Little's formula as

$$W = L/\lambda.$$

III. ESTIMATION

Clearly, the analysis of any problem along the lines indicated in this paper depends on the ability to obtain estimates of the parameters (μ_n, s_n) for the Erlangian density to be used. It turns out that maximum-likelihood procedures are not too difficult for this problem and would be recommended, especially in light of their desirable large-sample properties. An outline of this procedure follows. In the event of a small sample the reader is referred to [8].

The parameter s_n can assume only integer values and for a given value of s_n, it can easily be shown that the maximum-likelihood estimator for μ_n is simply $s_n/\bar{t}$, $\bar{t}$ the sample mean. This follows from the fact that the density may be written as

$$b_n(t) = \mu_n^{s_n} t^{s_n - 1} e^{-\mu_n t} / (s_n - 1)!$$

and hence the log-likelihood based on k observations as

$$\mathscr{L} = s_n k \ln \mu_n + (s_n - 1) \sum_{i=1}^{k} \ln t_i - \mu_n \sum_{i=1}^{k} t_i - k \ln (s_n - 1)!.$$

Therefore

$$\partial \mathscr{L} / \partial \mu_n \Bigg|_{\substack{\mu_n = \hat{\mu}_n, \\ s_n = \hat{s}_n}} = 0 = \hat{s}_n k / \hat{\mu}_n - \sum_{i=1}^{k} t_i$$

and

$$\hat{\mu}_n = \hat{s}_n/\bar{t}.$$

Now to get the complete pair $(\hat{\mu}_n, \hat{s}_n)$, consider s_n to be a continuous variable and proceed in the usual way to obtain the *MLE* of x for the gamma distribution

$$f(t) = \mu^x t^{x-1} e^{-\mu t}/\Gamma(x).$$

A description of this problem appears, for example, in a recent article by Choi and Wette [1]. The estimate $\hat{x}$ is found as the numerical solution to the nonlinear equation

$$k \log \hat{x} + k\gamma + k/\hat{x} - k\hat{x} \sum_{i=1}^{\infty} \{i(i+\hat{x})\}^{-1} = k \log \bar{t} - \sum_{i=1}^{k} \log t_i.$$

where γ is Euler's constant. So, finally, the *MLE* of s_n, $\hat{s}_n$, is either $[\hat{x}]$ or $[\hat{x}]+1$ (where $[x]$ is the greatest integer in x), depending which gives a higher value of the log-likelihood.

IV. CONCLUSION

Techniques of the sort outlined in this paper should have fairly wide applicability. One such possibility which comes right to mind is the continuous-review $(S-1, S)$ inventory problem [3]. If demand on an inventory system is Poisson, the reorder leadtime a random variable, and the policy selected $(S-1, S)$ with one-for-one ordering, then the orders enter a single-server queue and queuing results would be used to obtain expected inventory costs as a function of S in order to obtain the optimal value of S. If, in addition, the leadtimes have arbitrary state-dependent distribution functions, then the approach to state-dependent queuing described herein would be used to obtain the expected inventory costs.

REFERENCES

[1] Choi, S. C. and R. Wette, "Maximum Likelihood Estimation of the Parameters of the Gamma Distribution and Their Bias," Technometrics *11*, 683–690 (1969).

[2] Crabill, T. B., "Sufficient Conditions for Positive Recurrence and Recurrence of Specially Structured Markow Chains," Opns. Res. *16*, 858–867 (1968).

[3] Gross, D. and C. M. Harris (1970). "On One-for-One Ordering Inventory Policies with State-Dependent Leadtimes" (to appear in Operations Research).

[4] Harris, C. M. (1967). "Queues with State-Dependent Stochastic Service Rates," Opns. Res. *15*, 117–130 (1967).

[5] Kendall, D. G., "Some Recent Work and Further Problems in the Theory of Queues," Theory of Prob. and Its Applic. *9*, 1–13 (1964).

[6] Kotiah, T. C. T., J. W. Thompson, and W. A. O. Waugh, "Use of Erlangain Distributions for Single-Server Queuing Systems," J. Appl. Prob. *6*, 584–593 (1969).

[7] Rosenshine, M., "Queues with State-Dependent Service Times," Transpn. Res. *1*, 97–104 (1967).

[8] Shenton, L. R. and K. O. Bowman (1970). "Small Sample Properties for the Gamma Distribution (abstract)." Ann. of Math. Stat. *41*, 1384–1385 (1970).

ASYMPTOTIC INFERENCE ABOUT A DENSITY FUNCTION AT AN END OF ITS RANGE*

Lionel Weiss

Cornell University

ABSTRACT

For each n, $X_1(n), \ldots, X_n(n)$ are independent and identically distributed random variables, with common probability density function

$$f(x) = 0 \text{ for } x < \theta$$

$$f(x) = c(x-\theta)^{\alpha}[1 + r(x-\theta)] \text{ for } x \geq \theta,$$

where c, θ, α, and $r(y)$ are all unknown. It is shown that we can make asymptotic inferences about c, θ, and α, when $r(y)$ satisfies mild conditions.

1. INTRODUCTION

For each n, $X_1(n), \ldots, X_n(n)$ are independent and identically distributed random variables, with common probability density (with respect to Lebesgue measure) $f(x)$, distribution function $F(x)$, satisfying

$$f(x) = 0 \text{ for } x < \theta$$

$$f(x) = c(x-\theta)^{\alpha}[1 + r(x-\theta)] \text{ for } x \geq \theta,$$

where c, θ, α, $r(y)$ are all unknown, except that we know $c > 0$, $\alpha > -1$, and $|r(y)| \leq Ky^{\gamma}$ for all y in some interval $[0, \Delta]$, where K, γ, and Δ are all positive, but are otherwise unknown. A very great variety of density functions which assign zero probability to the left of some value θ satisfy these conditions. For example, $f(x) = w(x)[1 + r(x-\theta)]$, where $w(x)$ is a Weibull density with location parameter θ.

We are interested in making inferences about c, θ, and α, without making any stronger assumptions about $r(y)$ than those already made. It seems clear that we will have to use only the smaller observations, or else the particular unknown $r(y)$ will influence our inference about c, θ, and α, and we will lose control over levels of significance and confidence coefficients. But we would like to use as many of the smaller observations as possible, to avoid wasting information. The main purpose of this paper is to investigate how many of the smaller observations we can use, as n increases.

2. THE ASYMPTOTIC DISTRIBUTION OF THE SMALLEST ORDER STATISTICS

Let $Y_1(n) \leq Y_2(n) \leq \ldots \leq Y_n(n)$ denote the ordered values of $X_1(n), \ldots, X_n(n)$. For each n, let $k(n)$ denote a positive integer not greater than n. Define $Q_i(n)$ as $\frac{cn}{\alpha+1}(Y_i(n) - \theta)^{\alpha+1}$ for $i = 1, \ldots, k(n)$. Denote the joint probability density function for $Q_1(n), \ldots, Q_{k(n)}$ by $g_n(q_1,$

*Research supported by NSF Grant GP 21184.

$\ldots, q_{k(n)})$. The standard formula gives that $g_n(q_1, \ldots, q_{k(n)})$ is equal to

$$\frac{n!}{(n-k(n))!}\left[\frac{1}{\alpha+1}\left(\frac{\alpha+1}{cn}\right)^{\frac{1}{\alpha+1}}\right]^{k(n)}\left[1-F\left(\theta+\left\{\frac{(\alpha+1)q_{k(n)}}{cn}\right\}^{\frac{1}{\alpha+1}}\right)\right]^{n-k(n)}$$

$$\times\prod_{i=1}^{k(n)}\left\{q_i^{-\frac{\alpha}{\alpha+1}}f\left(\theta+\left\{\frac{(\alpha+1)q_i}{cn}\right\}^{\frac{1}{\alpha+1}}\right)\right\}.$$

if $0<q_1<\ldots<q_{k(n)}$, and $g_n(q_1, \ldots, q_{k(n)})$ is equal to zero otherwise.

Let $U_1, \ldots, U_{k(n)}$ be independent and identically distributed random variables, each with density function e^{-u} for $u>0$, zero for $u<0$. Define Z_i as $U_1+\ldots+U_i$, for $i=1, \ldots, k(n)$. The joint probability density function for $Z_1, \ldots, Z_{k(n)}$, which we denote by $h_n(z_1, \ldots, z_{k(n)})$, is equal to $e^{-z_{k(n)}}$ if $0<z_1<\ldots<z_{k(n)}$, and is equal to zero otherwise.

In this section, we prove the following:

THEOREM: If $\lim_{n\to\infty} k(n)=\infty$, $\lim_{n\to\infty}\frac{k(n)}{n^\delta}=0$ for every $\delta>0$, and if for each n, G_n is any Lebesgue measurable region in $k(n)$-dimensional space, then

$$\lim_{n\to\infty}\left|\int_{G_n}\cdots\int g_n(z_1, \ldots, z_{k(n)})dz_1\ldots dz_{k(n)}-\int_{G_n}\cdots\int h_n(z_1, \ldots, z_{k(n)})dz_1\ldots dz_{k(n)}\right|=0.$$

PROOF: First we investigate $F(y)$ in a neighborhood of θ. Of course, $F(y)=0$ if $y\leq\theta$. If $y>\theta$, we have

$$F(y)=\int_\theta^y[c(x-\theta)^\alpha+c(x-\theta)r(x-\theta)]dx \tag{2.1}$$

$$=\int_0^{y-\theta}[ct^\alpha+ct^\alpha r(t)]dt$$

$$=\frac{c(y-\theta)^{\alpha+1}}{\alpha+1}+c\bar{r}(y-\theta)\int_0^{y-\theta}t^\alpha dt$$

$$=\frac{c(y-\theta)^{\alpha+1}}{\alpha+1}[1+\bar{r}(y-\theta)],$$

where

$$\inf_{0\leq x\leq y-\theta} r(x)\leq\bar{r}(y-\theta)\leq\sup_{0\leq x\leq y-\theta} r(x)$$

and therefore

$$|\bar{r}(y-\theta)|\leq K(y-\theta)^\gamma \text{ for } 0\leq y-\theta\leq\Delta.$$

Now, by using the expression for $F(y)$ (given by (2.1)), when $0<Z_1<\ldots<Z_{k(n)}$ we can write $\log\frac{g_n(Z_1, \ldots, Z_{k(n)})}{h_n(Z_1, \ldots, Z_{k(n)})}$ as the sum of the following four expressions:

$$\sum_{i=0}^{k(n)-1}\log\left(\frac{n-i}{n}\right). \tag{2.2}$$

$$\sum_{i=1}^{k(n)}\log\left[1+r\left(\left\{\frac{(\alpha+1)Z_i}{cn}\right\}^{\frac{1}{\alpha+1}}\right)\right]. \tag{2.3}$$

$$(n-k(n))\log\left[1-\frac{Z_{k(n)}}{n}\left\{1+\bar{r}\left[\left(\frac{\alpha+1}{cn}Z_{k(n)}\right)^{\frac{1}{\alpha+1}}\right]\right\}\right]. \tag{2.4}$$

$$Z_{k(n)}. \tag{2.5}$$

In absolute value, expression (2.2) is less than $k(n)\left|\log\left(\frac{n-k(n)+1}{n}\right)\right|$. Using the formula $\log(1-x)=-\frac{x}{1-\bar{x}}$, where $|\bar{x}|<|x|$, we find that in absolute value expression (2.2) is less than $k(n)\left|\frac{\frac{k(n)-1}{n}}{1-\epsilon(n)}\right|$, where $|\epsilon(n)|<\frac{k(n)-1}{n}$. Since by assumption $\frac{k(n)}{\sqrt{n}}$ approaches zero as n increases, it follows that (2.2) converges to zero as n increases.

Define W_n as $\frac{Z_{k(n)}-k(n)}{\sqrt{k(n)}}$. The asymptotic distribution of W_n is standard normal. $\frac{Z_{k(n)}}{n}=\frac{k(n)}{n}+\frac{\sqrt{k(n)}\,W_n}{n}$, and thus $\frac{Z_{k(n)}}{n}$ converges stochastically to zero as n increases This implies that with probability approaching one as n increases, expression (2.3) is less in absolute value than

$$k(n)\log\left(1+K\left|\frac{(\alpha+1)Z_{k(n)}}{cn}\right|^{\frac{\gamma}{\alpha+1}}\right)=k(n)\log\left(1+K\left|\frac{(\alpha+1)k(n)}{cn}+\frac{(\alpha+1)}{cn}\sqrt{k(n)}W_n\right|^{\frac{\gamma}{\alpha+1}}\right),$$

and the last expression is easily seen to converge stochastically to zero as n increases, using the fact that $\lim_{n\to\infty}\frac{k(n)}{n^\delta}=0$ for any $\delta>0$. Thus (2.3) converges stochastically to zero as n increases.

Using the fact that $Z_{k(n)}=k(n)+\sqrt{k(n)}\,W_n$, the property of $\bar{r}(y)$ developed above, and the convergence of $\frac{k(n)}{n^\delta}$ to zero, the expansion of the log in (2.4) shows that (2.4) can be written as $-Z_{k(n)}+\Delta(n)$, where $\Delta(n)$ converges stochastically to zero as n increases.

Collecting the information about (2.2), (2.3), and (2.4) developed above, and taking (2.5) into account, we have shown that $\log\frac{g_n(Z_1,\ldots,Z_{k(n)})}{h_n(Z_1,\ldots,Z_{k(n)})}$ converges stochastically to zero as n increases. The proof of the theorem is now completed by using the argument given on pages 261–262 of Ref. [1].

In the next section, we discuss the application of the theorem to inference about c, θ, and α. The application is based on the fact that the asymptotic distribution of $Y_1(n),\ldots,Y_{k(n)}(n)$ does not depend on $r(y)$.

3. APPLICATION TO LARGE-SAMPLE INFERENCE

There are many ways to choose a sequence $\{k(n)\}$ satisfying the hypotheses $\lim_{n\to\infty}k(n)=\infty$, $\lim_{n\to\infty}\frac{k(n)}{n^\delta}=0$ for every $\delta>0$. For one example, $k(n)=$ the largest integer in $\log n$. For another example, $k(n)=$ the largest integer in $n^{(\log n)^{-1/2}}$. Suppose that we arbitrarily choose one particular sequence $\{k(n)\}$ satisfying $\lim_{n\to\infty}k(n)=\infty$, $\lim_{n\to\infty}\frac{k(n)}{n^\delta}=0$ for every $\delta>0$. Then the theorem of Section 2 tells us

that for all asymptotic probability calculations, we can assume that

$$Y_i(n) = \theta + \left[\left(\frac{\alpha+1}{cn}\right)\left(U_1 + \ . \ . \ . \ + U_i\right)\right]^{\frac{1}{\alpha+1}}$$

for $i = 1, \ldots, k(n)$, where $U_1, U_2, \ldots$ are independent and identically distributed, each with density function e^{-u} for $u > 0$. We note that the unknown function $r(y)$ plays no role in the joint distribution of $Y_1(n), \ldots, Y_{k(n)}(n)$ resulting from the assumption that

$$Y_i(n) = \theta + \left[\left(\frac{\alpha+1}{cn}\right)\left(U_1 + \ . \ . \ . \ + U_i\right)\right]^{\frac{1}{\alpha+1}},$$

so our objective of getting rid of $r(y)$ asymptotically has been achieved.

Even using only $Y_1(n), \ldots, Y_{k(n)}(n)$ for purposes of inference about c, θ, and α, and thus getting rid of $r(y)$ asymptotically, the problem of constructing asymptotically optimal tests or estimates is very difficult, being closely related to the corresponding inference problems with respect to three unknown parameters (location, scale, and shape) of a Weibull distribution. However, using the representation of $Y_1(n), \ldots, Y_{k(n)}(n)$ in terms of $U_1, \ldots, U_{k(n)}$, it is easy to construct consistent estimators of c, θ, and α, as we now show.

Suppose $k(n)$ is even for each n. We can write

$$\frac{Y_{\frac{k(n)}{2}} - Y_1}{Y_{k(n)} - Y_1} \xrightarrow{L} \frac{\left[\frac{1}{k(n)}\left(U_1 + \ . \ . \ . + U_{\frac{k(n)}{2}}\right)\right]^{\frac{1}{\alpha+1}} - \left(\frac{U_1}{k(n)}\right)^{\frac{1}{\alpha+1}}}{\left[\frac{1}{k(n)}\left(U_1 + \ . \ . \ . + U_{k(n)}\right)\right]^{\frac{1}{\alpha+1}} - \left(\frac{U_1}{k(n)}\right)^{\frac{1}{\alpha+1}}},$$

and the expression on the right is easily seen to converge stochastically to $2^{-\frac{1}{\alpha+1}}$ as n increases. It follows directly that a consistent estimator of α is

$$\frac{\log 2}{\log\left[\dfrac{Y_{k(n)} - Y_1}{Y_{\frac{k(n)}{2}} - Y_1}\right]} - 1 = \hat{\alpha}(n), \text{ say.}$$

Also, assuming that $k(n)$ is chosen so that $\dfrac{\log n}{\sqrt{k(n)}}$ approaches zero as n increases, it can be shown that

$$\frac{(\hat{\alpha}(n) + 1)k(n)}{n(Y_{k(n)} - Y_1(n))^{\hat{\alpha}(n)+1}}$$

is a consistent estimate of c. $Y_1(n)$ is a consistent estimate of θ. Once again, we emphasize that these estimates are consistent, but not asymptotically efficient.

The same analysis can be applied to the situation where $f(x)$ has an upper endpoint, or both an upper and a lower endpoint.

REFERENCE

[1] L. Weiss, "The Asymptotic Joint Distribution of an Increasing Number of Sample Quantiles," Annals of the Institute of Statistical Mathematics 21, 257–263 (1969).

OPTIMAL POLICIES FOR A MULTI-ECHELON INVENTORY SYSTEM WITH DEMAND FORECASTS*

Donald L. Iglehart
Stanford University
and
Richard C. Morey
Decision Studies Group

ABSTRACT

Consider an inventory system consisting of two installations, the stocking point and the field. Each period two decisions must be made: how much to order from outside the system and how much to ship to the field. The first decision is made based on the total amounts of stock then at the two installations. Next a forecast of the demand in the current period is sent from the field to the stocking point. Based upon a knowledge of the joint distribution of the forecast and the true demand, and the amounts of stock at the two installations, a decision to ship a certain amount of stock to the field is taken. The goal is to make these two decisions so as to minimize the total n-period cost for the system. Following the factorization idea of Clark and Scarf (1960), the optimal n period ordering and shipping policy, taking into account the accuracy of the demand forecasts, can be derived so as to make the calculation comparable to those required by two single installations.

1. INTRODUCTION AND SUMMARY

We consider an inventory system consisting of two installations in series. All incoming orders arrive at the upper installation (the stocking point) and all demands originate at the lower installation (the field). Each period two decisions are made at the stocking point: how much to order from outside the system and how much to ship to the field. The decision of how much to order from the outside is made first based on the amounts of stock on hand at the two installations. If any stock is ordered, it is delivered immediately to the stocking point. Next a forecast of the demand in the current period is sent from the field to the stocking point. Based on the forecast and the amounts of stock at the two installations, a decision to ship a certain amount of stock to the field is taken. This stock is shipped to the field immediately. Our goal is to make these two decisions in such a manner as to minimize the total n-period cost for this inventory system.

There is an alternative interpretation for the demand forecast. In actual inventory systems the field installation usually transmits its current demand to the stocking point in the form of a requisition. Frequently the requisitions received at the stocking point are inaccurate due to key punch errors, transmission errors, or errors in preparation. The net effect of these errors is that the stocking point may process a demand considerably different from the true demand. These errors lead to errors in the knowledge of the asset positions at the two installations. If these potential errors in asset position are corrected at the beginning of each period, then we can think of the errored demand transmitted from the field as being our demand forecast and the model will be the same. If the errors are not corrected, we are led to models of a different sort which we treat in Ref. [3]. For a slightly different treatment of imperfect demand information, see Ref. [5].

Our cost structure is as follows. At the stocking point (installation 1) we assume a linear purchasing

*This research was supported by the Office of Naval Research Contract Nonr-4457(00), NR-347-001.

cost with a set-up cost for any positive order and a general convex one-period expected holding and shortage cost. In the field (installation 2) we assume a linear cost for shipping from the stocking point and a general convex one-period expected holding and shortage cost conditional on the demand forecast. Excess demand in the field is completely backlogged.

With this cost structure we find that the optimal n-period ordering policy at the stocking point is of the (s_n, S_n) form and the optimal shipping policy of the single critical number type. Furthermore, to calculate these policies we can take advantage of the fact that the total optimal expected cost for the n-period problem can be factored in such a manner as to make the calculations comparable to those required for two single installation models.

The idea of the factorization for multi-echelon models was first introduced by Clark and Scarf [1] in a highly original paper. Our model is essentially theirs with the addition of the demand forecast. This trick of factorization has also been exploited in Clark and Scarf [2].

2. THE OPTIMAL ORDERING AND SHIPPING POLICIES

We begin by setting up the functional equations governing the inventory system. Let the purchase cost at the stocking point be given by

$$c(z)=\begin{cases} K+c_1\cdot z, & z>0 \\ 0, & z=0, \end{cases}$$

where z is the amount ordered. Let $L_1(y_1)$ be the one-period expected holding and shortage cost at installation 1 when the total system stock after the current order has arrived is y_1. The shipping cost from installation 1 to installation 2 is linear with unit cost c_2. The one-period expected holding and shortage cost at installation 2 is $L_2(y_2|\eta)$, where y_2 is the inventory level in the field after receipt of any stock shipped and η is the demand forecast for the period. Let $f_n(x_1, x_2)$ be the optimal expected n-period cost when the total system stock at the beginning of the first period is x_1, and the inventory level at installation 2 is x_2. Similarly we let $g_n(x_1, x_2, \eta)$ be the optimal n-period cost when the total system stock after delivery of the order placed at the beginning of the first period is x_1, the inventory level at installation 2 is x_2, and the demand forecast is η. If $\alpha(0<\alpha\leq 1)$ is the discount factor and ξ is the demand in the first period, then the sequences $\{f_n : n\geq 0\}$ and $\{g_n : n\geq 1\}$ satisfy the following system of functional equations:

$$g_n(x_1, x_2, \eta)=\min_{x_2\leq y_2\leq x_1}\{c_2\cdot(y_2-x_2)+L_2(y_2|\eta)+\alpha E_{\xi|\eta}[f_{n-1}(x_1-\xi, y_2-\xi)]\},$$

and

$$f_n(x_1, x_2)=\min_{y_1\geq x_1}\{c_1(y_1-x_1)+L_1(y_1)+E_\eta[g_n(y_1, x_2, \eta)]\},$$

where $n\geq 1$, $f_0\equiv 0$, $E_{\xi|\eta}$ denotes the conditional expectation with respect to the demand ξ given the forecast η, and E_η denotes the expectation with respect to the forecast η.

The principal tool used in obtaining our main result is the following lemma of Karush [4].

LEMMA 1: *Let* G(y) *be a real valued convex function on* $(-\infty, +\infty)$ *and form the function*

$$F(x_1, x_2)=\min_{x_1\leq y\leq x_2}\{G(y)\}.$$

Then

$$F(x_1, x_2)=G^1(x_1)+G^2(x_2),$$

where G^1 *is convex nondecreasing and* G^2 *is convex nonincreasing*. By using this lemma, we easily obtain Theorem 1:

THEOREM 1: *If* L_1 *is convex and* $L_2(\cdot|\eta)$ *is convex for all values of* η, *then*

$$g_n(x_1, x_2, \eta) = g_n^1(x_1, \eta) + g_n^2(x_2, \eta), \tag{1}$$

and

$$f_n(x_1, x_2) = f_n^1(x_1) + f_n^2(x_2) \tag{2}$$

for $n \geq 1$, *where* f_n^2 *and* $g_n^2(\cdot, \eta)$ *are convex for all* η, *and* f_n^1 *and* $g_n^1(\cdot, \eta)$ *are K-convex for all* η.

PROOF: The proof will be by induction on n. Let $n=1$. Then

$$g_1(x_1, x_2, \eta) = \min_{x_2 \leq y_2 \leq x_1} \{c_2 \cdot y_2 + L_2(y_2|\eta)\} - c_2 \cdot x_2.$$

Using Lemma 1 we have (1) with both $g_1^1(\cdot, \eta)$ and $g_1^2(\cdot, \eta)$ convex for all η. Now f_1 can be written as

$$f_1(x_1, x_2) = \min_{y_1 \geq x_1} \{c_1(y_1 - x_1) + L_1(y_1) + E_\eta[g_1^1(y_1, \eta)]\} + E_\eta g_1^2(x_2, \eta). \tag{3}$$

But the first term on the right-hand side of (3) is K-convex by the standard theory, cf. Scarf [1], and the second term is clearly convex. Thus we have (2) for $n=1$.

Assume now that the result is true for $n=N-1$. Then

$$g_N(x_1, x_2, \eta) = \min_{x_2 \leq y_2 \leq x_1} \{c_2 \cdot y_2 + L_2(y_2|\eta) + \alpha E_{\xi|\eta}[f_{N-1}^2(y_2 - \xi)]\} + \alpha E_{\xi|\eta}[f_{N-1}^1(x_1 - \xi)]. \tag{4}$$

Again the first term on the right-hand side of (4) is the sum of a convex function of x_1 and a convex function of x_2. The second term is K-convex. Since the sum of a convex function and a K-convex function is K-convex, (1) follows for $n=N$. To complete the proof we must show (2) for $n=N$. But this follows by the same argument used for $n=1$.

As an immediate corollary we obtain the form of the optimal ordering and shipping policies.

COROLLARY 1: *Under the hypothesis of Theorem 1, the optimal ordering policy in the n-period is of* (s_n, S_n) *type:*

if $x_1 \leq s_n$, *order* $(S_n - x_1)$,

if $x_1 > s_n$, *order nothing;*

the optimal shipping decision is of $\bar{x}_n(\eta)$ *type: for a forecast of* η

if $x \leq s_n$, *and* $x_2 \leq \bar{x}_n(\eta)$, *ship* $\min\{(\bar{x}_n(\eta) - x_2), (S_n - x_2)\}$.

if $x_1 > s_n$, *and* $x_2 \leq \bar{x}_n(\eta)$, *ship* $\min\{(\bar{x}_n(\eta) - x_2), (x_1 - x_2)\}$.

if $x_2 > \bar{x}_n(\eta)$, *ship nothing.*

Furthermore, the policy parameters s_n, S_n, *and* $\bar{x}_n(\eta)$ *can all be calculated recursively based on cost functions of only one variable.*

3. ACKNOWLEDGEMENT

We are grateful to Professor A. F. Veinott, Jr. of Stanford University for helpful discussions on this model.

REFERENCES

[1] Clark, A. and H. Scarf. "Optimal Policies for a Multi-Echelon Inventory Problem." Management Sci. *6*, 475–490 (1960).

[2] Clark, A., and H. Scarf. "Approximate Solutions To a Simple Multi-Echelon Inventory Problem," chap. 5 in *Studies in Applied Probability and Management Science*, edited by K. Arrow, S. Karlin, and H. Scarf. (Stanford University Press, Stanford, California, 1962).

[3] Iglehart, D. and R. Morey. "Inventory Systems with Imperfect Asset Information (forthcoming).

[4] Karush, W. "A Theory in Convex Programming," Nav. Res. Log. Quart. *6*, 245–260 (1957).

[5] Morey, R. "Inventory Systems with Imperfect Demand Information" (to appear in this journal).

[6] Scarf, H. (1960). "The Optimality of (s, S) Policies in the Dynamic Inventory Problem." chap. 13 in *Mathematical Methods in the Social Sciences*, edited by K. Arrow, S. Karlin, and P. Suppes (Stanford University Press, Stanford, California, 1960).

SIMULATION OF MULTI-ECHELON MACRO-INVENTORY POLICIES

Sheldon E. Haber

The George Washington University
School of Engineering and Applied Science
Institute for Management Science and Engineering

ABSTRACT

Alternative repair part inventory policies are examined for a multi-echelon logistics system. The policies assessed pertain to the design of multi-echelon systems rather than the evaluation of stock level decisions for individual repair parts. Although the context is one of a military inventory system, the policies examined are of importance in nonmilitary systems where item failure rates are low, and only periodic access to resupply is possible.

INTRODUCTION

In this paper several alternative repair part inventory policies are examined within the context of a multi-echelon logistics system. The vehicle for this study is a multi-echelon simulator. The simulator relates demand and supply for individual repair parts, but its primary utility is in assessing what may be termed macro-inventory policies, i.e., inventory policies pertaining to the overall structure of an inventory system. The particular inventory system modeled is the logistics support system for the Polaris submarine force. The context is one of a military inventory system: however, the policies examined are of importance in nonmilitary systems where item failure rates are low and only periodic access to resupply is possible.

The terminology, macro-inventory policies, is used to emphasize the main thrust of the paper which is to examine policies pertaining to the design of multi-echelon support systems rather than the assessment of stock level decisions for any particular repair part. For example, one of the more important decisions in the design of an inventory system is the choice of the number of echelons defining the system. Generally, it is only after this decision is made that attention is given to the problem of determining stock levels for individual line items. In terms of overall cost and effectiveness of an inventory system, the former problem area may be the more important one.

Although we are concerned with the design of multi-echelon logistics systems, in this paper an extant structure is studied for which real-life data were available. For the most part, the types of data used here are not available in the design stage of an inventory system. Even so, it can be instructive to use an extant inventory system for the purpose of improving the state of the art of structuring macro-inventory policies.

Based on the simulation experiments described in this paper, two conclusions appear warranted. First to maintain a high degree of effectiveness in a multi-echelon logistics system, a sufficient amount of stock should be placed with forward echelons, i.e., the ultimate consumer units, to meet uncertain demands. Second, given adequate stock levels at forward echelons, inventory policies can be imple-

mented which reduce investment in stock at resupply echelons and which may not greatly reduce overall system effectiveness. The inventory policies that we examine illustrate these conclusions.

The simulation model used consists of a three echelon logistics system having nine submarines, one tender, and one depot. The purpose of the simulator is to provide measures of the cost and effectiveness of alternative inventory policies for the logistics system as an entity. The emphasis on the word "alternative" should be noted. Because of the difficulty of simulating the actual logistics environment for a complex system, one is constrained in drawing conclusions regarding the cost and effectiveness of a single policy. This constraint is significantly reduced when two or more alternative sets of policies are compared. For the latter objective, a reasonably realistic simulation environment and a well-defined procedure for assessing the results of the simulation are sufficient.

The outline of the paper is as follows. A brief description of the simulator is given in section 1. The inventory policies for which simulations were performed are described in section 2. In section 3, we address the problem, common to all simulations, of assessing the cost-effectiveness of the policies simulated.

1. THE MULTI-ECHELON SIMULATOR

The main features of the simulator have been discussed in detail in [4]. The particular features described therein are the physical system, the accounting system, the generation of random events, simulator operating modes, and simulation input files. In this section, we summarize some of the important characteristics embodied in the simulator.

As noted above, the simulator models a three echelon system consisting of nine submarines, one tender, and one depot. The lowest echelon contains the nine submarines which are the ultimate consumers of the system. However, because submarines have their own storage capability, they are also suppliers of logistics support. Periodically submarines go on patrol and during each patrol simulated, random failures are generated for each repair part. Failed units are replaced from submarine stock by the ship's crew as long as replacement units are available. As failed units are replaced, a submarine's on-hand inventory is reduced by the number of units replaced. When replacement units are no longer available, further failures result in shortages of material which are recorded and used as a proxy measure of logistics system effectiveness.

Upon completion of a patrol, each submarine is refitted by the tender which constitutes the second echelon of support. During the refit period, a submarine is supplied with repair parts that it has used while on patrol and on-site repairs of failed items are made. Outstanding demands for repair parts which were not filled in previous refit periods are also met if sufficient stock is available. As a result of these actions, submarine stocks are replenished and tender stocks are simultaneously diminished. When tender stock is insufficient to meet a submarine's requirements, the tender initiates an order on the depot. If material is received from the depot and if there is still an insufficiency of stock, a submarine departs for the next patrol with less units of the item in stock than desired, i.e., with less than the item's allowed quantity or some or all units of the item in a failed condition.

The third support level in the simulation is the depot. Depot stocks are reduced when shipments are made to the tender and increased when repair parts are received from production and/or repair facilities. Depending on whether an item is a reparable or consumable, and if the former, the extent to which it can be economically repaired, a portion of the units ordered by the depot is treated as if it were processed through repair shops while the remainder is treated as if it were newly produced from production facilities. The time needed to augment item stock levels at the depot thus depends on an

item's repair cycle time and/or production lead time. In the simulation, the return time for an item is modified by random shocks to permit the possibility of early arrival and, as more often the case, late arrival of material at the depot. All demands not satisfied on first pass by the depot are backlogged and met at a later date.

For the simulations evaluated in this paper, two modes of transportation—normal air and priority air—are possible for resupplying tenders.* Except where otherwise noted, priority air transportation is used to transfer high essentiality items between the depot and tender.† In one of the inventory policies evaluated, the use of priority transportation for high essentiality items resulted in excessive transportation costs. The policy was then changed to require the use of normal air transportation for all shipments to the tender. Transportation costs were reduced, but logistics system effectiveness was reduced as well. The magnitude of these reductions is described in section 3.

For each of the two modes of transportation, there is a preassigned probability that replacement units requested for a submarine will arrive after the submarine has departed on its next patrol. For priority air the probability of late arrival was estimated to be 0.056, for normal air 0.172. To facilitate the accounting for material within the simulation model, it is assumed that in the event material arrives late, it becomes available to all consumers on a first-come, first-served basis. It is also assumed that transportation resources for resupply of material are adequate for all relevant inventory policies. Two reasons support this assumption. Repair items are typically small in size and only a small number of different repair parts are shipped from a depot at any one time.

The simulation model of the Polaris logistics system is a much simplified version of the real life system. Although in practice no two ships are configured in exactly the same manner, it has been assumed that the nine submarines do have the same installed population. More importantly, the real life system consists of 2 depots, 5 tenders, and 41 submarines, the first depot supporting one tender and the second depot supporting the remaining four tenders.

Despite the limited size of the logistics system modeled by the simulator, it replicates several important features of the real life system. As indicated below, for each item at each echelon, the probability of a shortage depends only on the item's characteristics, e.g., its military worth and unit price, and is independent of the number of stations at the next lowest echelon. Thus, item probabilities of stockout for the real life system would be in the same neighborhood as those for the system modeled by the simulator. On the cost side, real life system costs can be approximated by a linear function of costs for the simulated system.

Information pertaining to the characteristics of the repair parts carried in the Polaris logistics system was provided through the use of Navy files containing data on military essentiality, unit price, repair cycle time, etc.** Additional data were also obtained from which the probability of late arrival of material at the tender could be computed. With these important aspects of the real world incorporated in the simulation model, it was judged to be sufficiently realistic for the purpose at hand.

---

*Resupply of tenders by sea is also possible, but this alternative was eliminated on the basis of earlier simulations not reported here.

† A discussion of the Polaris military essentiality system is found in [1].

**The total number of different wearable parts for a Polaris submarine is approximately 35,000. Because of the large amounts of time that would have been needed to perform simulation experiments on the entire population of parts, they were limited to a sample of items. The sample contained 8,763 different parts, all of the items managed by the Strategic Systems Projects Office. Of all the highest essentiality items in the entire population, 59 percent were in the sample. Among high essentiality items, including those of the highest essentiality, 66 percent were in the sample.

Mention needs to be made of the usage rates used in generating submarine demands for repair parts. Expected item usage rates per patrol were developed from technician's estimates contained in the Navy files just referenced. For consistency, these patrol usage rates were used in both the computation of item stock levels at each echelon and in the simulation of units demanded per item per patrol.† Since our purpose was to estimate the effectiveness of alternative inventory policies, rather than to measure effectiveness for the real life system, the technician's estimates were used without modification in assessing the adequacy of supply to meet demand.

2. ALTERNATIVE INVENTORY POLICIES

In this section we describe several alternative inventory policies. In choosing the policies to be simulated, the objectives sought were twofold: (1) to assess the effects on a Polaris-type logistics system of various rules for stocking the tender echelon, and (2) to assess the effects of a reduction in stock levels at the end-consumer, i.e., submarine, echelon. The decision to focus on these aspects of multi-echelon macro-inventory policy was based on the structure of the Polaris logistics system. As noted previously, for the Polaris logistics system, resupply of submarines is periodic. While submarines are on station, they have no access to the resupply system. This characteristic of the logistics system suggests that consistent with dollar and storage constraints, the range and depth of items, i.e., the number of different parts stocked and number of units stocked, respectively, at the submarine echelon should be as large as possible. On the other hand, the basic tree-structure of the Polaris logistics system and the uncertainties of demand suggest that the depot should be considered as the primary resupply echelon. Given these premises, we examined the impact on logistics system effectiveness of policies whose intent was to reduce system costs by reducing stock levels, first at the tender echelon and then at the submarine echelon. No simulations examining the effect of reduced depot stock levels on system effectiveness were possible during the time available for the study. Some implications of the analysis for this aspect of multi-echelon inventory policy are made at the end of the paper.

A multi-echelon policy, denoted as the "standard" policy, was chosen against which alternative policies could be compared. The particular policy chosen as the standard was, with one exception,* the policy defined by [6]. It would be beyond the scope of this paper to describe the rules for determining item stock levels for submarine, tender, and depot echelons as found in detail in [6]. It is sufficient to note that in the standard policy, the tender is loaded with safety level stock, economic-add-on stock (to reduce the number of times material is shipped to the tender), and endurance stock. Moreover, in the standard policy, the tender is given parity with the depot as a resupply point. This parity is indicated not so much by the dollar investment in stock at the tender vis-a-vis the depot, but by the philosophy underlying the way in which the tender is stocked.

In the standard policy, the tender is regarded as a mobile, mini-depot. While, in practice, tenders are anchored close to shore, they have the capability of operating at sea.‡ The objective of stocking the tender as a mini-depot is achieved in [6] by establishing the same item stockout probabilities at the tender as at the depot, and by providing the tender with endurance stock in addition to safety level

†Examination of actual submarine usage data indicates that the average number of different repair parts used per patrol over the entire population of parts was approximately equal to the number generated per patrol by the simulator for the parts contained in the sample.

*See footnote on page 122.

‡The opportunity to exercise this capability depends on strategic considerations, the evaluation of which must obviously be left to others.

stock. In the alternative tender policies examined here, the role of the tender is quite different from the one just described.

In all inventory policies including the standard policy, stock levels at each echelon are computed using a model which is a generalization of the model applied to the "newsboy problem" (see [2]). This model provides optimal item stock quantities for a single echelon, but when it is used sequentially at each echelon in a logistics system, the overall distribution of stocks within the system need not be optimal. It is important to note here that all other things being equal, the model computes the probability of stockout for an item as a function of its military essentiality, unit price, and unit cube.† Thus, the probability of stockout for an item at a given echelon is independent of the number of stations at the next lower echelon.

Whereas the standard policy just described stocks a large number of units at the tender, in the next two policies no stock is placed at the second echelon.

In one policy the tender is treated as a loading platform, but it still retains its capability of operating at sea. In this policy, denoted by "tender as a loading platform," the tender is maintained as an integral part of the logistics system, but no investment is made in tender stock. All demands placed on the tender are satisfied directly from depot stock. The tender thus serves as a temporary storage point requesting material from the depot in direct response to demands by end consumers. In general, the tender as a loading platform policy will lead to increased transportation costs resulting from the substitution of priority transportation for the normal mode of transfer to insure timely arrival of material to the end consumer. But even with the use of priority transportation, there will be some reduction in submarine post-refit stock levels and submarine effectiveness vis-a-vis the case where material is prepositioned at the tender. By use of the simulator, changes in transportation and other costs and logistics system effectiveness, implicit to this policy, are estimated.

A very different approach is taken in a second policy. The logistics system is treated as if it consisted of only two echelons—submarines and a depot. In practice, however, an advanced land base situated in the same general area where the tender is normally stationed would be needed to perform minor repair work and assist in the installation of repaired components aboard submarines. It is assumed that an advanced land base exists,* and that no stock is kept at this base so it too may be considered as a loading platform. The difference between this policy, denoted as the "no tender" policy, and the previous one is the loss of the capability of mobility associated with the tender. Additionally, one might expect some increase in trans-shipment time between depot and submarines either because the loading of material aboard submarines will require more time in the absence of the tender and/or some repair work normally done at the tender will be shifted backwards to the depot. To take account of the possibility of increased trans-shipment time, it is assumed in the no tender policy that material cannot be furnished to submarines during the refit period in which they request resupply. This last assumption is equivalent to assuming that the probability of late arrival of material in the forward staging area is 1.00.‡ Since material always arrives late, only normal air transportation is used in this policy.

It has been suggested that where the logistics system is restricted to peridic resupply of end consumers, stock levels at the lowest echelon should be as large as possible within the available dollar

†The latter variable is used only at the submarine echelon where a space constraint must be met.

*In what follows questions concerning the political feasibility of this policy are ignored.

‡In practice, some percentage of items, particularly consumable items, should be deliverable to submarines during the refit period. This assumption, therefore, leads to an overstatement of the loss of effectiveness associated with the no tender policy.

and space constraints in order to maintain a high degree of effectiveness.† That is to say, when it is important that no item, among several thousand items, lacks replacement units in order to avert a malfunction in a larger system, in this case, the submarine, it may be necessary to stock each item at this echelon in such quantities that the probability of experiencing a shortage is extremely small. In the context of the present discussion, an example of a contrary policy would be to reduce initial investment in stock at the submarine echelon rather than at the tender echelon. In particular, a contrary policy might assume the acceptability of some minimum probability of stockout for any item, e.g., one shortage of a repair part per 100 patrols. This policy is used in [5] and in unpublished Navy memoranda describing echelon stock level computations. It can be reformulated as a policy which imposes a maximum repair part protection level of 0.99 and prohibits stocking of units which raise the protection level for any item above 0.99. This policy, which we denote as the "maximum protection level, 0.99" policy, is assessed here. To maintain comparability, we assume that this policy is the same as the standard policy except that item protection levels at each echelon are constrained to a maximum value of 0.99. Assessment of this policy provides a means of evaluating the proposition that centralization of stocks at higher echelons, achieved by reducing stocks of end consumer units, can be non-optimal when immediate resupply of the latter is not possible.

The policies described above are defined explicitly in Table 1 in terms of initial dollar investment in stock for each echelon in the logistics system. In estimating initial investment in stock, calculations were first made for the simulated logistics system of nine submarines, one tender, and one depot.* These figures were then used as a basis for extrapolating to a representation of the Polaris logistics system comprising a force of 45 submarines supported by five tenders and one depot. The extrapolated figures (shown in Table 1) were obtained by multiplying the original cost data for each echelon of the simulated logistics system by five. The use of a linear function for estimating initial investment costs at the first two echelons is reasonable since the proxy system contains five times as many submarines

TABLE 1. *Initial Dollar Investment in Stock*[a]

Policies	Submarine	Tender	Depot	Total system
1. Standard	121	37	83	241
2. Tender as a Loading Platform	121	0	83	204
3. No Tender	121	0	83	204
4. Maximum Protection Level, 0.99	94	33	76	203

[a] In millions of dollars per 5-year period for a logistics system of 45 submarines, 5 tenders, and 1 depot.

†This proposition does not address the question of the mix of items to be stocked aboard submarines, i.e., should submarines be stocked with a large number of low priced items or a smaller number of items containing a larger proportion of high priced items? This question requires additional analysis.

*For all policies, item stock quantities at the depot consist only of safety level stock. This limitation on depot stock is the only difference between the standard policy and the one defined by [6]. The change in the standard policy was made to accentuate the "shock" on the logistics system of changes in inventory policy. For the policies considered, the effect is to overestimate transportation and depot order costs and to underestimate the effectiveness of the Polaris logistics system.

and tenders as the simulated one. At the depot echelon, this procedure introduces an element of difficulty. On the one hand, because of uncertainty of demand, depot stock requirements to support five tenders will be less than five times the requirement to support one tender. On the other hand, the Polaris logistics system contains two depots with some duplication of stock between them. Given the focus of our study, the estimating procedure used here seems adequate.

As can be seen from Table 1, the total initial investment in stock for the standard policy is larger by approximately 18 percent than for the other inventory policies. In terms of costs associated with initial investment in stock, the tender as a loading platform and the no tender policies are equivalent. But other costs associated with the construction, maintenance, and operation of tenders which are present in the former policy are absent in the latter one. Finally, we note the different distributions of stock between Policy 2 (and 3) and Policy 4, although the total initial investment in stock over all echelons is almost the same.

The inventory policies defined in Table 1 are but several of a large number of multi-echelon policies which could have been simulated. The tender as a loading platform and no tender policies represent limiting cases of a "light" tender load. However, it may well be that reductions in investment in stock, if reasonable at all, should be made at the submarine echelon.† In the next section, figures are presented which suggest significant reductions in logistics system effectiveness may result from a reduction in stock at the submarine echelon, but this may not be the case if the reduction is made at the tender echelon.

3. COST-EFFECTIVENESS EVALUATION

In any simulation, the problem of measuring cost and effectiveness is as troublesome, if not more so, as the problem of design of the simulation. In the present context, the major difficulties pertain to the measurement of effectiveness of a military logistics system and the translation of changes in effectiveness into units which permit comparison with changes in costs. In general, the problems of measuring effectiveness outweigh those pertaining to the measurement of costs. They are particularly difficult to handle when there is no market mechanism for measuring effectiveness. In this section, we first present estimates of operating costs and measurements of system effectiveness, and then attempt a reconciliation of the two sides of the cost-effectiveness relationship.

Estimates of operating costs, based on the simulation model, are shown in Table 2. These are computed over a 5-year period which is taken as the average duration between fundamental equipment design changes for Polaris submarines. As in the case of the previous table, all figures are computed by multiplying the cost estimates for the simulated system by a factor of five.

The least troublesome of the costs in Table 2, at least conceptually, are those relating to transportation, depot order, repair, and production costs since for the policies considered these consist only of operating expenses. But even here one encounters the difficulty of obtaining data from which reasonable cost estimates can be made. For example, no information could be found for the cost of repairing failed items. In this case, we assumed that repair cost was a fraction—one-quarter—of the cost of production. As a proxy for the latter cost, the unit price of the item as found in the Navy files was used. The average depot order cost was estimated from [3] as $50 per initial order and $25* per follow-on

†As noted earlier, no simulations based on reduced investment in stock at the depot echelon were performed so this aspect of macro-inventory policy could not be assessed.

*Order costs per item ranged from $24 for purchase orders processed by buying at retail outlets to $55 for purchases procured through advertised contracts. The average cost for all types of purchases was $28. See Table VI-B in [3].

TABLE 2. *Operating Costs*[a]

	Policies[b]				
	1	2A	2B	3	4
Inventory	107.1	90.8	88.6	88.2	87.9
Transportation	2.2	68.3	5.5	3.8	3.0
Depot Order	9.9	24.4	21.6	16.2	10.2
Repair	56.0	56.5	55.3	53.8	56.1
Production	103.2	104.4	102.4	101.2	102.7
Total	278.4	344.4	273.4	263.2	259.9

[a] In millions of dollars per 5-year period for a logistics system of 45 submarines, 5 tenders, and 1 depot.

[b] Policies:
1. Standard.
2A. Tender as a loading platform—priority air transportation for all highly essential parts.
2B. Tender as a loading platform—normal air transportation for all parts.
3. No tender.
4. Maximum protection level, 0.99.

order.† The cost of normal air transportation was estimated at $8 per item delivered to the tender. This estimate was based on tariff schedules of the Material Air Transport System and tonnage and transaction data supplied by the Strategic Systems Projects Office. Data on the cost of priority air transportation, however, were unavailable. It was suggested to the author that $100 per item delivered to the tender would be a reasonable estimate for priority air delivery, and this was the assumption used here.

One component of operating costs associated with transportation is handling charges. We have made no attempt to estimate incremental handling charges at the depot resulting from reduced investment in stock at the tender. In effect, we considered this element of cost to be zero. The reasons for taking this approach are twofold. First, the extra manpower required at the depot can be offset to some extent by a reduction in manpower at the tender (or advanced land base). Second, by reducing the volume of material to be stored at these latter sites, it may be possible to alter their physical configuration, thereby effecting a reduction in investment in plant. Although it is not clear that there would be a complete cancellation of costs, we proceed on this assumption.

In estimating operating costs, conceptual problems, as well as data problems, are encountered. For example, should cost of inventory be measured in terms of initial investment in stock or in some other manner. We estimated inventory costs by measuring the cost of material consumed and assuming an allowance for the return of unused material after an accounting period, which in our case covers 5 years. The amount allowed would depend on actual accounting practices and on a variety of factors, the most important being the rate at which material becomes obsolescent. For the sophisticated Polaris system, we assumed that one-half of unused stock (based on figures provided by the simulator) could be recovered after 5 years.

†A follow-on order is defined as a modification to an initial order made prior to the delivery date established for the initial order. Any additional procurement initiated after the delivery date is counted as another initial order.

Our approach to the empirical and conceptual problems is typical. In the absence of the required information, a number of assumptions have been made whose only justification is that they do not seem unreasonable. Should the reader have more information, he can modify the estimates in Table 2 and reevaluate our conclusions.

Several observations can be made concerning the figures in Table 2. Looking at the structure of operating costs, production and repair costs account for half or more of operating costs,* but they vary very little among policies. The latter finding is not unexpected since these costs are related only to consumption and are independent of the total investment or distribution of initial investment in stock.

Perhaps the most surprising figure in Table 2 is the cost of transportation for the tender as a loading platform policy (Policy 2A) when priority air transportation is used to resupply all highly essential items. As can be seen from Table 3, the extremely high cost of transportation for this policy is due to two factors: the large number of shipments from the depot due to the zero stock levels at the tender and the heavy use of priority air transportation due to the large proportion of highly essential items in the sample.†

TABLE 3. *Number of Shipments to Tender by Mode of Transportation and Number of Depot Orders by Type*[a]

	Policies[b]				
	1	2A	2B	3	4
Shipments to Tender:					
Normal Air	54,653	25,469	138,406	95,099	55,248
Priority Air	18	134,498	0	0	1,537
Total	54,671	159,967	138,406	95,099	56,785
Depot Orders:					
Initial	24,598	34,804	34,583	34,548	25,027
Follow-On	29,942	124,787	103,442	60,539	31,602
Total	54,540	159,591	138,025	95,087	56,629

[a] Per 5-year period for a logistics system of 45 submarines, 5 tenders, and 1 depot.
[b] See Table 2 for policy definitions.

The effect of the former factor is also seen on depot ordering cost. Since depot stocks were purposely limited to only safety level stock, one-for-one ordering occurs at the depot when a demand is placed on the tender. The point of interest here is the strong interrelationship between initial investment in stock and the components of operating cost. These relations are often overlooked, not because they are not known, but because of the difficulty of obtaining estimates of the variables and parameters influencing operating costs.

---

*Here again it should be recalled that what is being estimated are not real life production and repair costs, but their counterparts based on the expected failure rates used to compute echelon stock quantities.

†The figures for transportation and depot order costs in Table 2 are derived from Table 3 by using the cost factors cited in the text.

To reduce the total operating cost associated with the tender as a loading platform policy when priority air transportation was used for all highly essential items, an alternative version, Policy 2B, was simulated. In Policy 2B, normal air transportation is used for all shipments between depot and tender. Transportation cost for this new policy is still higher than for the standard policy, but it is sufficiently reduced to bring total operating cost below that for the standard policy.

For the no tender policy, one notes that total operating cost is even less than for Policy 2B, despite the fact that normal air transportation is used for all items in both policies. More than half the difference in total operating cost between these policies is accounted for by depot order cost which is a function of the number of requests for stock placed on the depot. The explanation for the reduction in requests for stock under the no tender policy is found in the construction of the simulation model. As mentioned earlier, when material assigned to a submarine arrives after the submarine has departed for its next patrol, it becomes available to other submarines on a first-come, first-served basis. When more than one customer is given the units assigned to another, batching of requests occur reducing shipments to the advanced land base and (with one-for-one ordering at the depot) depot orders to production and repair facilities. For the no tender policy, where it was assumed that material always arrived late, batching of requests occurred frequently. This happened also, but less often in Policy 2B where the probability of late arrival was much smaller.*

The rule that material ordered by one customer may be given to other customers is, itself, an example of a macro-inventory policy. From the simulation experiments, it would appear that this rule would be an effective one in reducing operating costs. Logistics system effectiveness may also be reduced, however, if the material given to a second customer is more urgently needed by the first customer. The probability of this event occurring can be minimized if "trading" of material is prohibited when the need of the first customer is urgent, e.g., when material is needed by the first customer to replace failed units of a repair part as contrasted to the case where the stock level for an item is positive but below the allowed quantity. A similar rule for trading of stock may be applied in another context. Since more than one submarine is in refit at one time at a given site, repair parts could be traded among submarines. In general, it should not be too difficult to establish rules by which the implementation of trading results in reduced operating costs and even an expected net gain in effectiveness.

Of all the policies simulated, operating cost was least for the maximum protection level policy. Not only was inventory cost less for this policy, but in addition transportation and depot order costs were kept low due to the carrying of stock at the tender.

Still to be discussed is the logistics system effectiveness associated with each policy. Before doing so, it is worth pointing out that with only a small investment in tender stock, demands on the depot can be substantially reduced for the tender as a loading platform policy, thereby lowering operating costs for this policy to a level much below that of the standard policy. This can be accomplished, for example, by stocking the tender with items having a low unit price and a high expected demand rate. This approach was employed in [6] in the computation of economic-add-on stock using the formulation

$$\text{Economic-add-on stock} = 10\sqrt{U_i/P}. \tag{1}$$

*It should be noted that the ratio of the number of shipments to the tender (or advanced base) to the number of depot orders will, in general, not equal 1, as might be thought for the case where depot stock for an item consists only of safety level stock. For example, the filling of back-orders could result in more than one shipment for a single depot order. This event occurred when depot stock was positive, but insufficient to supply all the units requested for a customer. On the other hand, depot orders were not always matched by shipments. If depot stock for an item were zero units, multiple requests for the item resulted in multiple depot orders, but only a single back-order shipment. As may be seen from Table 3, total shipments and total depot orders were approximately equal.

where U_t is the expected monthly usage of an item at the tender, and P is the item's unit price. In (1), an item's stock level is higher the higher its expected monthly usage and the lower its unit price. As can be seen from Table 4, for the standard policy, the stock placed at the tender using (1) comprises most of the range but only a fraction of the total initial investment in stock. Thus, for a minimal investment of $0.44 million, orders on the depot can be substantially reduced. The cost of stocking the depot as a mini-depot is substantially higher, i.e., $37.0 million. Of particular interest, too, is that the economic-add-on stock occupies a space of 354 cuft, i.e., a space which measures only a little more than 7 ft in each dimension.

TABLE 4. *Initial Investment in Tender Stock, Standard Policy*

	Total investment	Investment in economic-add-on stock
Range of Items Stocked[a]	8,663	5,226
Depth in Units of Items Stocked[a]	198,695	96,770
Space (cuft)[a]	3,987	354
Dollars (millions)[b]	37.0	0.44

[a] Per tender.
[b] Per 5-year period for five tenders.

Although a policy using (1) only to stock the tender was not simulated, it is clear that this policy which does not depart in principle from the tender as a loading platform policy would reduce operating cost and increase effectiveness. It might be added that a similar effect could be achieved if the no tender policy were modified to permit the placement of items with low unit price and high expected usage at advanced land sites.

To this point in the discussion, reference has been made to logistics system effectiveness, but no definition has been provided. Typically, logistics system effectiveness is measured at each echelon in the system. This approach yields a vector of measurements, which in itself is not undesirable, but such measurements can be misleading. For example, the end consumer may be able to supply an acceptable percentage of demands from his own stocks despite difficulties encountered by resupply points in satisfying his *initial* demands for replacement units. This case can arise where end consumers have sufficiently large stocks on hand and demand is relatively low. These two conditions characterize the Polaris system. To avoid misinterpretation, we define logistics system effectiveness solely in terms of submarine effectiveness, although even for this single echelon, a number of different effectiveness measurements are possible. The measures of effectures provided by the simulator for the submarine echelon are shown in Tables 5 and 6.

The most common measure of submarine (logistics) effectiveness is the first one shown in Table 5. In practice, supply effectiveness is defined as the percentage of demands for repair parts during a patrol for which submarine stock is sufficient to satisfy all units demanded. In this definition, a failure to supply demanded units is counted only during the patrol in which the demand is originally placed. Our definition departs from this in that during each patrol unfilled demands are treated as new transactions for which supply is less than demand. Thus, our definition of supply effectiveness yields a lower estimate than the one used in the field. It is of some interest, therefore, to note that submarine supply

effectiveness is insensitive to large changes in the level and distribution of stock in the logistics system. In particular, considering this measure only and neglecting the problem of translating changes in effectiveness into dollar terms, the maximum protection level policy compares rather favorably with the standard policy.

TABLE 5. *Submarine Effectiveness* [a]

	Policies [b]				
	1	2A	2B	3	4
Supply Effectiveness	0.997	0.996	0.995	0.990	0.977
Number of Different Parts with a Shortage:					
All Parts	3.0	3.9	4.7	9.5	20.5
Parts with Highest Essentiality	0	0.03	0.06	0.55	2.63

[a] Average patrol values for a 5-year period.
[b] See Table 2 for policy definitions.

In Table 5, we also show two other measures of submarine effectiveness. These measures pertain to the number of items with a shortage, i.e., cases where demand exceeded supply resulting in a non-operating part. The coefficient of variation for these measures is much larger than that for the supply effectiveness measure. It can be observed that for these measures, too, the standard policy yields an extremely high level of effectiveness. For this policy, the average number of different repair parts with a shortage, per patrol, was only 3.0 while no shortage was recorded over a 5-year period for any repair part with the highest essentiality. The level of effectiveness appears to fall somewhat for Policies 2A, 2B, and 3 and drops markedly for the last policy in which the probability of stockout is constrained to a minimum of 1 percent.

The differences in effectiveness among policies are portrayed more sharply in Table 6. The figures in this table show the percentage of patrols, in a 5-year period, with shortages of highest essentiality parts. The figure in the first column indicates that for Policy 1, no shortage of parts with highest essentiality occurred during any patrol. On the other hand, for Policy 4, no shortage for this class of parts occurred in only 6.7 percent of the patrols.

Table 6 raises at least two questions with which one must grapple in order to approach some of the conceptual difficulties in reconciling effectiveness and operating cost.

An immediate question is the number of shortages of highest essentiality items which cause a submarine to terminate patrol. If the answer were two, then no patrols would be aborted for Policies 1 and 2A while for Policies 2B, 3, and 4, the percentage of patrols which could not be completed would be 0.6, 11.7, and 71.7, respectively.*

Another question in interpreting the figures in Table 6 pertains to the timing of shortages. For example, assuming agreement that two shortages of items of highest essentiality reduce patrol effectiveness from 1.0 to 0, the level of effectiveness attained during a patrol will depend on whether the second shortage occurs on the first or last day of the patrol. If one assumes a uniform distribution of shortages

*A more sophisticated approach would be to estimate the probability of a patrol being terminated when 1, 2, 3, . . ., highest essentiality parts have a shortage. With these estimates, the expected percentage of patrols which cannot be completed because of insufficient supply support could be computed for each policy. The problem is much more difficult than portrayed here, however, since one would want to consider combinations of shortages of highest essentiality items with items of lesser essentiality.

TABLE 6. *Percent of Patrols With Shortages of Highest Essentiality Parts*

Number of Highest Essentiality Parts With a Shortage	Policies[a]				
	1	2A	2B	3	4
0	100.0	97.2	94.4	60.6	6.7
1		2.8	5.0	27.8	21.7
2			0.6	8.3	25.0
3				2.8	18.3
4				0.6	14.4
5					7.8
6					4.4
7					0.6
8					0.6
9					0.6
	100.0	100.0	100.0	100.0[b]	100.0[b]

[a] See Table 2 for policy definitions.
[b] Rounded to 100.0.

within any patrol, the expected loss of effectiveness would be one-half that indicated by the figures in Table 6.[†] Thus, for Policy 2B, 0.6 percent of the patrols would be terminated (assuming a critical value of two shortages), on average, at the mid-point of a patrol with a consequent expected reduction in effectiveness of 0.3 percent. This convention in computing expected loss of logistic system effectiveness is adopted below.

The problem of translating loss in logistics system effectiveness into dollar terms is admittedly even more difficult than the initial problem of measuring loss in effectiveness. Yet, discussion of the former problem area is unavoidable. Our approach was to estimate the incremental outlay in dollars required to maintain logistics system effectiveness for an alternative policy at the same level achieved by the standard policy. To compute this incremental outlay, we used the following expression:

(2)
$$\text{Incremental Capital Outlay} = \left[\frac{1}{1-\Delta E} - 1\right] \cdot I = \frac{\Delta E}{1-\Delta E} \cdot I,$$

where ΔE is the reduction in logistics system effectiveness for the alternative policy vis-a-vis the standard one, and I the estimated capital investment in plant and equipment per accounting time period for the alternative policy. In applying (2), we note that if ΔE equals 1/2, the incremental capital outlay equals I. The incremental increase in capital outlay per accounting period is then compared with the incremental decrease in [illegible]erating costs per accounting period. If the former is less than the latter, the performance of the alternative policy is judged to be better than that of the standard.

Given the published figure of $100 million for construction of a Polaris submarine, we estimated the life-time cost of constructing and maintaining (including maintenance of personnel) a Polaris submarine and Polaris tender at $200 million and $400 million, respectively. Thus for a force of 45 submarines and 5 tenders, total capital investment would be $11 billion, or $2.2 billion per 5-year period assuming an economic life for the system of 25 years. On the basis of these estimates, the incremental capital outlay for a 1 percent reduction in logistics system effectiveness would be $22.2 million.

[†] The same result is obtained if patrol effectiveness declines at the rate $1/n$ during a patrol, where n is the number of shortages which if incurred leads to termination of a patrol.

As noted above, the expected loss in logistics system effectiveness depends on the assumption made regarding the number of shortages of highest essentiality items that lead to termination of a patrol. For example, if a patrol is terminated when shortages are incurred for two or more items with the highest military essentiality, the expected loss of effectiveness for the tender as a loading platform policy (Policy 2B) is 0.3 percent, and the incremental capital outlay per 5-year period required to offset this loss is $6.6 million (see Table 7). From Table 2, one notes that the incremental decrease in operating cost (per 5-year period) for Policy 2B is $5.0 million. It should be recalled, however, that the latter figure may be substantially increased by a trivial dollar investment in items with low unit price and high expected demand at the tender echelon. Assuming that the shortage of two or more items of highest essentiality is needed to terminate a patrol, it would appear that the tender as a loading platform policy compares quite favorably with the standard policy.

TABLE 7. *Estimated Incremental Capital Outlay as a Function of the Critical Number of Highest Essentiality Parts Needed to Terminate a Patrol* [a]

Critical number	Policy [b]			
	2A	2B	3	4
1	31.2	63.4	467.6	1927.6
2	0	6.6	118.0	1229.4
3	0	0	32.9	670.1
4	0	0	5.7	364.1
5	0	0	−300.0	165.7
6	0	0	−300.0	70.4
7	0	0	−300.0	20.0
8	0	0	−300.0	13.2
9	0	0	−300.0	6.6
10 or more	0	0	−300.0	0.0

[a] In millions of dollars per 5-year period for a logistics system of 45 submarines, 5 tenders and 1 depot.
[b] See Table 2 for policy definitions.

The estimated incremental capital outlay per 5-year period for alternative assumptions regarding the number of highest essentiality items leading to termination of a patrol is shown in Table 7. In computing the figures for the no tender policy, Policy 3, the life-time costs used are $200 million for a Polaris submarine and $100 million for construction and maintenance of a repair capability at an advanced land base. The negative figures indicate the *reduction* in capital outlays for this policy (vis-a-vis the standard policy) when logistics system effectiveness is equal to that for the standard policy. Comparing the figures in Table 7 with the incremental operating costs for each policy, we find that Policy 3 ranks higher than the standard if the critical number of shortages is four or more. The same is true for Policy 4, but only if the critical number of shortages is eight or more. Note, here the savings are restricted to operating costs whereas in Policy 3 there is also a reduction in capital outlays. It is interesting to note that whereas Policy 2A ranks lower than the standard for all assumptions because of its higher operating cost and lower effectiveness, the former ranks higher than Policy 4 if the critical number of shortages is between 1 and 5. In general, Policy 4 ranks lower than the others if the critical number of shortages is small.

Notwithstanding the difficulties of estimating loss of effectiveness and converting this loss into a monetary equivalent, the data in Tables 6 and 7 suggest the following tentative conclusions. The standard policy, in which submarines are stocked with as wide a range and depth of items as possible consistent with dollar and space constraints, provides a very high level of logistics system effectiveness. For this reason, improvements in effectiveness from policies which would place still greater emphasis on the tender or depot as a stock point would be difficult to achieve and also expensive. On the other hand, although substantial reductions in investment in stock can be realized by placing less stock at the end-consumer echelon, this policy is likely to result in a marked reduction in logistics system effectiveness. Finally, the tender as a loading policy and no tender policies appear to rate favorably with the standard policy under not too restrictive assumptions regarding the number of shortages of highest essentiality items needed to terminate a patrol.*

These conclusions rest on two basic characteristics of the Polaris logistics system—resupply is periodic (only once after each patrol) and item usage rates, even when accumulated over the entire system, are low. The latter characteristic is of special importance in explaining the impressively small number of shortages for the tender as a loading platform and no tender policies. The effect of low demand rates on logistics system effectiveness can be illustrated by the following example. Assume that submarine and depot stock levels for an item are such that the probability of supply exceeding demand is 0.997 and 0.993, respectively, when the stocks at each echelon are considered separately. Then the probability of incurring a shortage of stock over the entire supply system during a two-patrol period covering 6 months, assuming that submarine stock is the only stock available during the first patrol and that depot stock is the only stock available for the second patrol, will be approximately 0.010. Since for most items, the return time from production and/or repair facilities is 6 months or less, this means that stock at the end-consumer and depot echelons *alone* would be sufficient to cover 99 out of 100 demands during the period in which material can be replenished throughout the multi-echelon inventory system.

The characteristic of low usage rates suggests that for a large proportion of items, stock need be kept only at two echelons and for a not unimportant class of items one echelon may suffice. As noted, we have concluded that in order to maintain a high level of logistics system effectiveness, item stocks should be placed, in general, at the end-consumer echelon. For items that are bulky and/or expensive but which also have the characteristic of high essentiality, the lowest echelon of support may be the tender or advanced land base. Given the policy of stocking an item at the submarine or tender depending on its cost and its military essentiality (or at both echelons if it is included among the items for which economic-add-on stock is computed), the primary resupply responsibility falls to the depot. But all items need not be stocked at the depot. If an item is expensive and has low military essentiality, but most important, if its usage rate is low, it probably can be deleted from the depot stock list with only a trivial effect on logistics system effectiveness. Because of the low failure rate, there is a high probability that another unit can be obtained from the manufacturer before the supply system experiences another demand for the item. And even if this is not the case, if the item is of low essentiality, the expected decline in system effectiveness should be small.

In concluding this paper, a final observation should be noted. Although a large number of assumptions were required throughout the analysis, an explicit effort was made to be conservative in the choice

*It should be noted that if the limiting assumption that all shortages occur at the beginning of a patrol is made, Policy 2B compares less favorably and Policy 4 much less favorably with the standard policy. The general conclusions, however, would remain unchanged.

Best Available Copy

of the sample, in structuring the simulator, and in estimating both operating costs and incremental capital outlays to offset loss of effectiveness. In all policies, one-for-one ordering was imposed at the depot echelon, thereby increasing depot order costs; the impact, however, was primarily on the two policies for which stock was eliminated at the middle echelon. Additionally, the extreme version of the tender as a loading platform policy and the assumption of late delivery for the no tender policy results in an over-estimate of loss in effectiveness for these policies. Because of the complexity of cost-effectiveness analysis, it is somewhat humbling to realize that when all is said and done, evaluation of the findings presented here ultimately depends on the individual reader's expertise, both in terms of information at hand and experience in the operating characteristics of complex systems.

REFERENCES

[1] Denicoff, M., J. Fennell, S. E. Haber, W. H. Marlow, F. W. Segel, and Henry Solomon, "The Polaris Military Essentiality System," Nav. Res. Logist. Quart. *11*, 235–257 (1964).

[2] Denicoff, M., J. Fennell, S. E. Haber, W. H. Marlow, and Henry Solomon, "A Polaris Logistics Model," Nav. Res. Logist. Quart. *11*, 259–272 (1964).

[3] Dunlap and Associates, Inc., *A Study of Procurement Costs at the Ships Parts Control Center, Vol. I, Concepts, Methodology and Results*, Stamford, Connecticut (1961).

[4] Haber, Sheldon E., "Simulation of a Multi-Echelon Logistics Support System," *Large Scale Provisioning Systems*, J. Ferrier, editor (English Universities Press Ltd., London, 1968), 483–504.

[5] U.S. Naval Supply Depot, "Inventory Control Manual, The Uniform Automated Data Processing System (UADPS), Part I, Requirements Determination for Consumables, ALRAND Rept 45, Mechanicsburg, Pennsylvania (12 Apr. 1965).

[6] U.S. Navy Special Projects Office, "Repair Parts Support Requirements for Special Projects Office Fleet Ballistic Missile Equipments," Instruction P4423.27A, Washington, D.C. (22 May 1967).

A NOTE ON A PAPER BY W. SZWARC

T. S. Arthanari and A. C. Mukhopadhyay

Indian Statistical Institute

ABSTRACT

In this note the authors call for a change of the optimality criteria given by Theorem 3 in section 5 of the paper of W. Szwarc "On Some Sequencing Problems" in NRLQ Vol. 15, No. 2 [2]. Further, two cases of the three machine problem, namely, (i) $\underset{k}{\text{Max}}\, A_k \leq \underset{k}{\text{Min}}\, B_k$ and (ii) $\underset{k}{\text{Max}}\, C_k \leq \underset{k}{\text{Min}}\, B_k$ are considered, and procedures for obtaining optimal sequences in these cases are given. In these cases the three-machine problem is solved by solving n (the number of jobs) two-machine problems.

INTRODUCTION

Section 5 of the paper of W. Szwarc "On Some Sequencing Problems" in NRLQ Vol. 15, 2; 1968, pp. 140–147 [2], deals with certain special cases of the well known Johnson's $3 \times n$ sequencing problem. The optimality criteria are subject to changes since several terms in L and R on page 142 are simply missing.

Conditions 2a and 3a of the theorem are considered alone and new procedures for obtaining optimal sequences in cases satisfying either of the conditions are developed. An example is given to illustrate the procedure.

Before discussing the problems involved we assume the reader to be familiar with [2]; however, no familiarity with [2] is required for section 2 of this note.

1. Corrected L and R

We restate the following notations from [2]:

$p = (1, 2, \ldots\ldots, n)$ denotes the sequence $1, 2, \ldots\ldots, n$ of the n jobs.

A_i, B_i, C_i denote the operating time of item i on machines A, B and C respectively.

Let

$$K_u = \sum_{i=1}^{u} A_i - \sum_{i=1}^{u-1} B_i \text{ and}$$

(1)

$$H_v = \sum_{i=1}^{v} B_i - \sum_{i=1}^{v-1} C_i \text{ for sequence } p.$$

For the sequence $p' = (1, \ldots\ldots, j-1, j+1, j, j+2, \ldots\ldots, n)$, denote by K'_u, H'_v the corresponding expressions similar to (1).

Let

(2) $$g(p) = \max_{1 \leq u \leq v \leq n} (H_v + K_u).$$

In addition, we introduce a notation

$$\bar{p} = (j_1, j_2, \ldots \ldots \ldots \ldots, j_n). \tag{3}$$

and $\bar{K}_u$, $\bar{H}_r$ the corresponding expressions similar to (1) for the sequence $\bar{p}$.

Now consider L and R given on page 142 of [2]. It is easy to see that the terms of H_r+K_u, $H'_r+K'_u$ for $u=j$, $j+1$ and $j+2 \leq v \leq n$ are missing. The correct expressions for L and R are given below.

$$L = \max\ (H_j+K_u,\ 1 \leq u \leq j;\ H_j+1+K_u,\ 1 \leq u \leq j+1;\ \underline{H_r+K_j,\ H_r+K_{j-1},\ j+2 \leq v \leq n})$$

$$R = \max\ (H'_j+K_u,\ 1 \leq u \leq j;\ H'_{j+1}+K'_u,\ 1 \leq u \leq j+1;\ \underline{H'_r+K'_j,\ H'_r+K'_{j+1},\ j+2 \leq v \leq n})$$

(the underlined expressions indicate the missing terms in L and R in [2].

This trivial error is crucial, however, since it leads to incorrect optimality criteria (see Theorem 3 on page 144 of [2]).

The correct form of L and R changes all three criteria presented in the mentioned theorem. For instance, criterion 1 will be as follows:

1. (a) B_j = constant, (b) $A_j < A_{j+1}$, (c) $C_j > C_{j+1}$ for all j.

2. Some Special Cases of the Three Machine Problem

Consider a sequence $\bar{p}$ defined by (3) and the corresponding $\bar{K}_u$ and $\bar{H}_r$, then,

$$g(\bar{p}) = \max_{1 \leq u \leq r \leq n} (\bar{H}_r + \bar{K}_u).$$

CASE I: Let $\operatorname{Max}_k A_k \leq \operatorname{Min}_k B_k$

then

$$\bar{K}_t \geq \bar{K}_{t+1} \text{ for } 1 \leq t \leq n-1$$

and

$$\begin{aligned} g(\bar{p}) &= \max_{1 \leq r \leq n} (\bar{H}_r + \bar{K}_1) \\ &= A_{j_1} + \max_{1 \leq r \leq n} \bar{H}_r \\ &= A_{j_1} + \max\ (B_{j_1},\ B_{j_1} + B_{j_2} - C_{j_1},\ \bar{H}_3, \ldots\ldots, \bar{H}_n) \\ &= A_{j_1} + B_{j_1} - C_{j_1} + \max\left[C_{j_1},\ \max_{2 \leq r \leq n}\left(\sum_{t=2}^{r} B_{j_t} - \sum_{t=2}^{r-1} C_{j_t}\right)\right]. \end{aligned}$$

Let P_i denote the set of all sequences with $j_1 = i$. We are interested in minimizing $g(\bar{p})$ over all possible sequences $\bar{p}$. For a fixed $j_1 = i$:

Let

$$I_i = \operatorname{Min}_{\bar{p} \in P_i}\left[\max_{2 \leq r \leq n}\left(\sum_{t=2}^{r} B_{j_t} - \sum_{t=2}^{r-1} C_{j_t}\right)\right].$$

I_i is the minimum idle time on the last machine C, for the two machine problem with machines B and C and the $(n-1)$ jobs other than i.

Hence, we can develop the following procedure to obtain an optimal sequence. For $i = 1, 2, \ldots\ldots\ldots\ldots, n$,

$$D_i = A_i + B_i - C_i + \max\ (C_i,\ I_i). \tag{5}$$

where I_i is as defined above. Let S_i be the corresponding optimal sequence.

$$\text{Find} \quad D_{i_0} = \min_i D_i.$$

Then an optimal sequence for the problem is given by (i_0, S_{i_0}).

Example:

Jobs	A_i	B_i	C_i
1	5	9	6
2	8	11	5
3	7	8	2
4	4	12	4

Here, we have, using Johnson's rule,

$$I_1 = 22 \text{ and } S_1 = (2, 4, 3)$$
$$I_2 = 19 \text{ and } S_2 = (1, 4, 3)$$
$$I_3 = 21 \text{ and } S_3 = (1, 2, 4)$$
$$I_4 = 17 \text{ and } S_4 = (1, 2, 3).$$

I_i can be found, using the expression for idle time on the last machine in the two machine problem or Gantt chart.

Now $A_i + B_i - C_i$ for $i = 1, 2, 3$ and 4 are 8, 14, 13 and 12, respectively.

As $I_i > C_i, \forall i$, we have, $D_1 = 30$, $D_2 = 33$, $D_3 = 34$ and $D_4 = 29$

Since D_4 is the minimum over D_i's, (4, 1, 2, 3) is an optimal sequence.

CASE II: Let $\max_k C_k \leq \min_k b_k$. Consider the sequence $\bar{p} = (j_1, j_2, \ldots j_n)$.

Then

$$\bar{H}_t \leq \bar{H}_{t-1}$$

and

$$g(\bar{p}) = b - c + (C_{j_n} + \max_{1 \leq u \leq n} \bar{K}_u) = (b-c) + [C_{j_n} + \operatorname*{Max}_{1 \leq u \leq n-1} (\bar{K}_u, a - b + B_{j_n})].$$

where

$$a = \sum_{i=1}^{n} A_i, \qquad b = \sum_{i=1}^{n} B_i, \qquad c = \sum_{i=1}^{n} C_i.$$

Similar to case I, fixing j_n one can develop the following procedure. For $i = 1, 2, \ldots \ldots \ldots n$. Calculate

$$D_i = C_i + \max (I_i, a - b + B_i)$$

when I_i is minimum idle time on the last machine B, for the two-machine problem with machines A and B and the $(n-1)$ jobs other than i. Let S_i be the corresponding optimal sequence.

$$\text{Find} \quad D_{i_0} = \operatorname*{Min}_i D_i.$$

Then an optimal sequence for the problem is given by (S_{i_0}, i_0).

ACKNOWLEDGEMENT

The authors are thankful to W. Szwarc for his useful suggestions.

REFERENCES

[1] Johnson, S. M., "Optimal Two and Three Stage Production Schedules with Setup Times Included," Nav. Res. Log. Quart., **1**, 61–68 (1954).

[2] Szwarc, W., "On Some Sequencing Problems," Nav. Res. Log. Quart., **15**, 127–155 (1968).

NEWS AND MEMORANDA

INTERNATIONAL CONFERENCE ON STOCHASTIC POINT PROCESSES

An International Conference on the topic "Stochastic Point Processes; Statistical Analysis. Theory and Applications" will be held at the IBM Research Center. Yorktown Heights. New York on August 2–7, 1971, the week before the International Statistical Institute meetings in Washington. D.C. The organizing committee consists of D. R. Cox and P. A. W. Lewis, Chairmen, M. S. Bartlett, J. Gani, K. Matthes, P. A. P. Moran, E. Parzen, R. Pyke, W. L. Smith, and D. Vere-Jones.

The aim of the conference is to bring together mathematicians and statisticians working in this field and workers in applied fields, such as ecology, neurophysiology, traffic studies, reliability, geography, forestry, epidemiology, and geophysics. Consequently, there will be three categories of papers presented at the conference:

i) Survey papers on the mathematical theory, statistical analysis, and models of univariate point processes, multivariate point processes, multidimensional point processes, and line processes;

ii) Review papers on the types of problems involving point processes encountered in fields of application such as ecology, neurophysiology, physics, forestry, reliability, traffic, geography, etc.

iii) A limited number of contributed papers on new work in the field.

We hope to have the survey papers available before the conference and also to print a compilation of open problems for discussion at the conference. Problems for inclusion should be limited to one typewritten page (8-1/2x11 in.) and be submitted before April 30, 1971. Each submission should include the author's name and address.

Papers presented at the conference will be published either in *Biometrika*, or in the *Journal of Applied Probability* and *Advances in Applied Probability*, subject to the usual acceptance and refereeing procedures.

Further information and submissions to the conference should be made to:

Dr. P. A. W. Lewis
Mathematical Sciences Department
IBM Research Center
P.O. Box 218
Yorktown Heights, N.Y. 10598, U.S.A.

☆ U.S. GOVERNMENT PRINTING OFFICE 1971 OL 414-155

INFORMATION FOR CONTRIBUTORS

The NAVAL RESEARCH LOGISTICS QUARTERLY is devoted to the dissemination of scientific information in logistics and will publish research and expository papers, including those in certain areas of mathematics, statistics, and economics, relevant to the over-all effort to improve the efficiency and effectiveness of logistics operations.

Manuscripts and other items for publication should be sent to The Managing Editor, NAVAL RESEARCH LOGISTICS QUARTERLY, Office of Naval Research, Arlington, Va. 22217. Each manuscript which is considered to be suitable material for the QUARTERLY is sent to one or more referees.

Manuscripts submitted for publication should be typewritten, double-spaced, and the author should retain a copy. Refereeing may be expedited if an extra copy of the manuscript is submitted with the original.

A short abstract (not over 400 words) should accompany each manuscript. This will appear at the head of the published paper in the QUARTERLY.

There is no authorization for compensation to authors for papers which have been accepted for publication. Authors will receive 250 reprints of their published papers.

Readers are invited to submit to the Managing Editor items of general interest in the field of logistics, for possible publication in the NEWS AND MEMORANDA or NOTES sections of the QUARTERLY.

CONTENTS

ARTICLES	Page
Tree-Search Algorithms for Quadratic Assignment Problems by J. F. Pierce and W. B. Crowston	1
Optimal Interdiction Policy for a Flow Network by P. M. Ghare, D. C. Montgomery and W. C. Turner	37
(0,1) Hyperbolic Programming Problems by P. Robillard	47
Application of the GLM Technique to a Production Planning Problem by J. P. Evans and F. J. Gould	59
Sequencing Many Jobs on a Multi-Purpose Facility by J. A. Buzacott and S. K. Dutta	75
Location of Facilities with Rectangular Distances Among Point and Area Destinations by G. O. Wesolowsky and R. F. Love	83
A Technique which Combines Modified Pattern Search Methods with Composite Designs and Polynomial Constraints to Solve Constrained Optimization Problems by B. F. Houston and R. A. Huffman	91
Finite Statistical Games and Linear Programming by R. A. Agnew and R. B. Hempley	99
On Queues with State-Dependent Erlang Service by C. M. Harris	103
Asymptotic Inference about a Density Function at an End of Its Range by L. Weiss	111
Optimal Policies for a Multi-Echelon Inventory System with Demand Forecasts by D. L. Iglehart and R. C. Morey	115
Simulation of Multi-Echelon Macro-Inventory Policies by S. E. Haber	119
A Note on a Paper by W. Szwarc by T. S. Arthanari and A. C. Mukhopadhyay	135
News & Memoranda	139